ISSN 0183-570X

DOSSIER HORS SERIE

L'ORDINATEUR INDIVIDUEL

CE DOSSIER PRÉSENTE UNE SÉLECTION DES MEILLEURS PROGRAMMES PUBLIÉS DANS L'ORDINATEUR INDIVIDUEL

50 PROGRAMMES BASIC

jeux, enseignement, professionnels, utilitaires, etc.

et un tableau comparatif de tous les Basic pour pouvoir utiliser les programmes sur :

Alice, Apple, Atari 400, Atom, BBC, Commodore 64, DAI, Dragon 32, Hector, Lynx, MZ 80 A, Oric, PET/CBM, TO 7, TI 99, TRS 80 mod. 1 et 3, Vic 20, ZX 81, ZX Spectrum

numéro spécial hors série n° 54 bis
Canada : 5.95$C-Belgique : 234FB-Suisse : 11FS 35 F

M 2946 - 54 bis - 35 FF

DES BONS POINTS POUR LA BASE DE DONNÉES dBASEII™ d'ASHTON TATE

dBASE II est un prodigieux outil de programmation qui vous permet de réaliser vos programmes de gestion beaucoup plus rapidement qu'avec des langages traditionnels.

Avec **dBASE II** vous créez vos grilles de saisie, ouvrez des fichiers relationnels, réalisez des tris et sélections, formatez vos listages à la demande. Vous réalisez ainsi en quelques jours des programmes se présentant comme ceux des grands systèmes et dont l'étude vous aurait demandé des mois.

dBASE II est interfaçable directement à des logiciels graphiques pour représentation en couleur à l'écran et imprimante ou table traçante des courbes, barres, camemberts relatifs au contenu de vos fichiers **dBASE II**.

dBASE II est disponible sous CPM/80, CPM/86 et MS-DOS. Les formats sont 8 pouces simple face/simple densité, visiotexte IBM, 5 pouces 1/4 · ordinateur personnel IBM, SIRIUS, ZENITH, DIGITAL, TEXAS, etc...

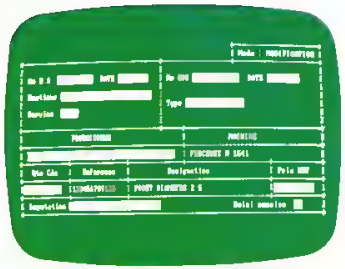
dBASE II est livré avec le manuel d'utilisation, un guide de formation et les messages d'écran en **Françeis**.

dBASE II est livré avec une disquette d'essai limitée à 15 enregistrements. Si sous 30 jours vous n'êtes toujours pas convaincu, nous vous rembourserons intégralement le produit.

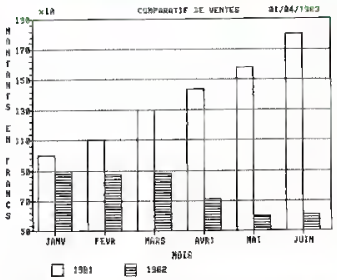
Le guide de formation de **dBASE II** est disponible séparément.

Ceux qui possèdent déjà la version Américaine peuvent se procurer la mise à jour française de **dBASE II**.

L'ÉCOLE PIGIER propose des stages de formation sur **dBASE II** (Tél. 233 44 88).
Contactez-nous pour de plus emples informations sur **dBASE II**.



Gestion de stock réalisée avec **dBASE II**



Graphique obtenu sur imprimante ordinaire avec le logiciel **dGRAPH** de **FOX & GELLER** ne nécessitant pas de fonction graphique du micro-ordinateur

POUR LA FRANCE

La Commande Electronique
7, RUE DES PRIAS — 27920 SAINT-PIERRE DE BAILLEUL
TÉL. (32) 52 54 02 TÉLEX LCE 180 855

Pour toutes précisions sur le société ou le produit présenté ci-dessus : Référence 101 du service-lecteurs (page 172)

Mettez l'informatique dans votre poche

L'ORDINATEUR DE POCHE est la seule revue française consacrée à l'informatique de poche.

Vous possédez déjà une calculatrice ou un ordinateur de poche programmable en BASIC. Vous trouverez dans L'ORDINATEUR DE POCHE mille astuces qui vous permettront de tirer un meilleur parti de votre machine.

Vous envisagez d'en acheter une, L'ORDINATEUR DE POCHE vous guidera dans votre choix et accompagnera vos premiers pas.

lisez l'Ordinateur de poche

16 Francs, chez votre marchand de journaux

BULLETIN D'ABONNEMENT A RETOURNER AUJOURD'HUI MEME A

L'ORDINATEUR DE POCHE, Service Abonnements,
59 RUE DE LA GRANGE AUX BELLES 75484 PARIS CEDEX 10

Nom _____ Prénom _____

Adresse _____

Pays _____ Code postal _____ Ville _____

Veuillez m'abonner pour 1 an à L'ORDINATEUR DE POCHE ;
Ci-joint mon règlement de 130 FF (Belgique : 1150 FB ; Suisse : 40 FS ; autres pays : 170 FF)
(Tarif par avion : Afrique francophone (sauf Zaïre) : 205 FF, autres Afrique, Amérique : 255 FF ; Asie, Océanie : 295 FF)

Référence 107 du service-lecteurs (page 172)

SOMMAIRE

UN INTERPRETE UNIVERSEL

15 **Parlez-moi Basic.** Nombre d'ordinateurs parlent Basic, et pourtant ils ne s'entendent pas : chacun parle son patois. Aussi nous vous proposons de passer à la loupe cent instructions sur dix-neuf ordinateurs parmi les plus répandus. Ce tableau comparatif vous permettra d'adapter facilement les programmes d'un ordinateur à l'autre.

JEUX

26 **Le Logotron : une nouvelle science** qui joue avec les mots. Rendez votre ordinateur individuel intarissable.

31 **Nous avons trouvé le monstre du Loch Ness !** Mais si ! Cette grande première française laisse le monde entier abassourdi. Voulez-vous jouer à le retrouver avec nous ?

32 **Jouez au morpion.** Un jeu injustement délaissé, mais après tout aussi beau que les échecs ou les dames, le go ou Othello.

35 **Le compte est bon.** Pour lancer cette petite phrase lors de l'émission télévisée bien connue.

36 **Le supporter de football.** Gardez en mémoire l'ensemble du championnat grâce à ce programme et à ses fichiers.

41 **Empilez des cubes :** les enfants résoudront ce problème bien plus vite que le programme que nous vous proposons, mais ce dernier présente bien des possibilités d'amélioration.

47 **Parcourez les méandres d'Isole :** un jeu simple à apprendre, mais gagnerez-vous contre votre ordinateur ?

51 **Le casse-brique** de votre ordinateur tient en quelques lignes : un jeu vidéo où vous aurez du mal à battre vos enfants.

53 **Donjons et dragons.** Découvrez les jeux de rôle avec ce programme qui vous fera frémir.

58 **Construisez des alexandrins** automatiquement grâce à ce programme. Tous ces vers n'auront pas un sens évident : comme les oracles, à vous de reconstituer le sens caché.

61 **Voulez-vous jouer au tennis ?** Votre ordinateur se charge de vous fournir le court, les balles et les raquettes.

64 **Voyage en Crète.** Perdu dans le labyrinthe, Thésée ne devait affronter qu'un seul minotaure. Ici vous serez seul, les minotaures seront légion et le labyrinthe invisible : un jeu terrifiant.

67 **Une course automobile** sur votre écran. Vous n'auriez pas réussi votre permis de conduire : en moins de 4 986 leçons.

71 **Lewis Caroll** ou comment la logique rejoint le merveilleux. Atmosphère, personnages et décor de cette aventure-jeu informatique sont authentiques, puisqu'ils sont humains.

75 **Le jeu d'Awele** est à l'Afrique ce que le Go est au Japon ou les échecs à l'Europe.

79 **Le I-Ching ou Livre des changements** correspond, dans la philosophie Zen, à nos oracles. Raffinons le procédé avec un ordinateur.

82 **L'alunissage est une manœuvre délicate.** Mais ici vous ne courrez aucun risque.

84 **Panique au fond des mers.** La torpille traverse sans dommage l'écran de votre ordinateur, et pourtant...

APPLICATIONS PROFESSIONNELLES

140 **Vérifiez vos chèques.** S'il vous arrive parfois de recevoir de nombreux chèques, un programme pour ne pas craindre les fautes de frappe lors de leur prise en compte.

142 **Calculez vos amortissements.** Déterminez les montants des amortissements sur les matériels que vous avez achetés.

146 **Les files d'attente,** un problème familier, quotidien presque. Pour les prévoir et s'organiser en conséquence. Ce programme de simulation prend pour exemple un péage d'autoroute. Il peut aussi s'appliquer à des situation telles que le passage aux caisses d'un supermarché, les transports de marchandises, la gestion de stocks, etc.

APPLICATIONS DOMESTIQUES

152 **Intéressez-vous à vos intérêts.** Un programme pour déterminer les taux d'intérêt que vous « offrent » les organismes financiers.

153 **La trésorerie familiale.** L'ordinateur excelle dans les calculs : il aide ici à gérer les dépenses et le budget de la famille.

157 **Un calendrier perpétuel.** Bien utile pour associer à chaque date le jour de la semaine correspondant.

159 **Le temps, c'est de l'argent.** Faites vous aider par votre ordinateur pour tenir votre agenda. Il a de la mémoire, lui.

161 **Pour organiser votre tournoi,** ne perdez pas la boule, même si vous avez de nombreux concurrents. Un programme à proposer à votre association sportive ?

162 **Autour de la corbeille.** Pour ceux qui maîtrisent bien ce qui se passe à la Bourse, gérez un portefeuille de titres.

Rédaction vente publicité

France et Etranger
39 rue de la Grange aux-Belles, 75484 Paris Cedex 10
Tél : (01) 238 66 10
Telex : LORDI 215 105F EDITEST

Belgique
Rédaction et publicité : 3 avenue de la Ferme Rose, B-1180 Bruxelles
Tél : (2) 345 99 10
Vente et abonnements : Soumillon 28 av. Massenet, B-1180 Bruxelles
Tél : (2) 344 19 23

Suisse : 27 route du Grand-Mont, CH-1052 Le Mont-sur Lausanne
Tél : (21) 32 61 77

L'ORDINATEUR INDIVIDUEL

DOSSIER PROGRAMMES N° 54 BIS DECEMBRE 83

UTILITAIRES

93 **Mettez de l'ordre dans vos données.** Vous devrez certainement à un moment ou à un autre, classer vos données, par exemple dans l'ordre alphabétique.

95 **Imprimer des étiquettes** est une tâche souvent fastidieuse. Pourquoi ne pas vous faire aider d'un ordinateur ?

98 **Un programme saisissant** qui vous permettra d'assurer la qualité des données que vous utiliserez.

101 **Gagnez de la place et du temps :** il vous suffit, pour cela, de faire varier la longueur de vos enregistrements.

105 **Pour gérer des fichiers,** il faut connaître la technique des listes linéaires doubles que vous présente ce programme.

111 **Découvrez les vertus de l'accès indexé.** Accès séquentiel ou accès direct ? L'accès indexé, troisième voie, combine les avantages des deux méthodes.

114 **Recherche séquentielle dans une table.** Une solution efficace si la rapidité n'est pas le critère le plus important.

115 **Recherche dichotomique.** Une méthode pour retrouver rapidement un élément dans une table ordonnée.

117 **Pour manipuler des caractères,** il est utile de disposer des six sous-programmes que nous vous présentons ici.

119 **Connaissez-vous les listes inversées ?** En complément à l'article de la page 111, pour en savoir plus sur l'accès indexé.

121 **Ne laissez pas entrer** n'importe quel octet dans vos fichiers. Deux sous-programmes pour entrer des nombres ou des caractères en les vérifiant à la saisie.

123 **Au commencement d'un programme** est le menu. Le mieux est de disposer d'un programme standard et de l'adapter.

124 **Réalisez la fusion ou l'intersection** de deux tables. Vous aurez besoin de ces méthodes pour comparer ou grouper deux fichiers.

125 **Pour avoir des programmes lisibles,** il suffit de demander à votre ordinateur de faire un effort de présentation.

127 **Accélérez vos recherches** en utilisant une table. En voici la recette.

128 **Evitez les destructions de fichier.** Une solution simple : un fichier différentiel.

ENSEIGNEMENT

129 **O tempora, o mores !** Ne voilà-t-il pas que les ordinateurs se mettent à aider l'apprentissage des déclinaisons ? Grâce à ce programme, (re)trouvez votre latin.

131 **Les mathématiques à l'école.** Les calculs de multiples et de diviseurs sont des opérations faciles : un programme utilisé par un enseignant pour ses cours.

133 **La multiplication en Egypte.** Il y a plus de deux mille ans, on utilisait déjà le système binaire. Apprenez à multiplier avec 0 et 1.

135 **Le calcul mental : un jeu d'enfant.** Pour vous essayer à un jeu ancien : la multiplication.

137 **Non ! Apprendre la géographie n'est plus rébarbatif** avec un ordinateur individuel.

MATHEMATIQUES

87 **Représentez vos données avec des histogrammes.** Un dessin vaut mieux qu'un long discours. Représentez vos données plutôt par des graphiques que par des chiffres.

90 **Tracer des courbes** sur écran : une solution visuelle pour études de fonctions.

Ce numéro spécial hors-série a été réalisé à partir d'articles déjà publiés dans les numéros antérieurs de *L'Ordinateur Individuel*.
Le lecteur ne manquera pas de remarquer que, durant sa longue vie (si, cinq ans déjà !), *L'Ordinateur Individuel* a évolué et que la maquette initiale a subi diverses modifications. Qu'il ne s'arrête pas à ces détails car *« qu'importe le flacon... »*.

ABONNEMENTS : PAGE 12
SERVICE LECTEURS : PAGE 172

Ce numéro contient en encart un bulletin d'abonnement et de cartes réponses pages 11 et 12.

LE CLUB DES CENT

•MAJUSCULE•

Le grand réseau national de micro-ordinateurs familiaux et consommables informatiques

Canon
commodore
SHARP
TEXAS INSTRUMENTS
THOMSON
SANYO
HEWLETT PACKARD
CASIO
ATARI

EN VENTE CHEZ :

02 ST QUENTIN
COGNET
21 rue Victor Hugo (23) 62 72 98

03 MOULINS
A. JOLY & FILS
Librairie Classique de Moulins
En face de la rue d'Allier
(70) 44 13 40
et 1 rue Denis Papin (70) 44 70 74

04 DIGNE
JEAN BIGARD
Place du Général de Gaulle (92) 31 11 53

05 GAP
DAVAGNIER
3 place Jean Mac Lellan (92) 51 01 17

06 NICE
SORBONNE PAPETERIE
33 rue de l'Hôtel
SORBONNE INFORMATIQUE
40 rue de Metz (93) 85 17 05
CANNES
SORBONNE PAPETERIE
7 rue des Belges (93) 39 91 13

09 FOIX
SURRE
28 rue Delcassé (61) 65 01 63

12 MILLAU
TREMOLET
8 place de Mendemas
B.P. 28 (65) 60 07 01

13 MARSEILLE
MAUPETIT
142 La Canebière (91) 48 21 77
MARSEILLE PAPETERIE
81 rue de Rome (91) 33 34 81

14 CAEN
LOISELET
2 10 place Pierre Fouche d (31) 85 56 96
LISIEUX
LOISON-BOBO
28 rue de la Libération (31) 31 13 11
CONDE SUR NOIREAU
LOISELET
5 rue St Pierre (33) 73 00 98

15 AURILLAC
DELPRAT
33 rue des Fenaisons (70) 48 42 18

16 ANGOULÊME
LHOMME
104 route de Bordeaux
(45) 95 27 37 (45) 95 63 51

17 LA ROCHELLE
SAINT MARTIN
16-18 rue St Yon B.P. 1007 (46) 41 24 40
Z.I. Périgny (46) 44 51 67

18 BOURGES
AU GRAND MEAULNES
Place Cujas (48) 24 05 71

VIERZON
PROUSTEAU
21 rue de la République
37 rue Victor Hugo (48) 75 25 17

21 CHATILLON-SUR-SEINE
MICHEL MAZZOTTI
Librairie-Papeterie
Châtillonnaise
3 rue du Président Carnot (80) 91 00 55

22 LOUDEAC
Mesdames CHEVE
3 avenue des Chemillonnes
B.P. 72 (95) 28 01 01

23 GUERET
SAINT-MARTIN JEUNE
11 Grande Rue (55) 52 41 36

24 SARLAT
G. BINET - J.L. AUBARRIER
45 rue de la République (53) 35 02 54

25 BESANCON
Ets CAMPONOVO
30 Grande Rue (81) 81 22 01
Dépôt rue Guy Lussac
Z.I. (81) 51 53 48

26 VALENCE
CRUSSOL
11 bd du Général de Gaulle (75) 43 00 35

28 CHARTRES
JEAN LEGUE
10 rue René Poiley (37) 21 17 17

29 BREST
Librairie JEAN JAURES
101 rue J. Jaurès (98) 44 61 35

30 NIMES
BUROTYP
21 rue de la République (91) 67 41 11

31 TOULOUSE
CASTELA S.A.
20 pl. du Capitole (61) 23 24 24

33 BORDEAUX
Ch. Robert
47-49 cours du Pas St-Georges (55) 52 11 13

34 MONTPELLIER
BORRAS-CLAVERIE
29 cours Gambetta (67) 58 11 54
SETE
BORRAS-CLAVERIE
25 quai Nord Bourgogne (67) 74 26 95

35 FOUGERES
MARY
10 av. du Général de Gaulle
(90) 99 75 40

38 GRENOBLE
UNIC IDESS
8 rue Anatole (76) 21 37 61
BOURGOIN JALLIEU
MAJOLIRE
anciens Ets FILLET
44 rue de la Liberté (74) 93 07 39

39 LONS LE SAUNIER
MAROUE MAILLARD
13 rue Lerouxte (84) 47 28 25

41 VENDOME
Ets PH. DENIS
26 et 58 av. Gare d'Yven (54) 77 02 57

42 SAINT ETIENNE
INTER BUREAU
SCHOOL LOISIRS
2 rue Foley (77) 23 02 03 +

44 NANTES
GRASLON
6 bis rue Duchie de Vanse Illes
(40) 74 15 50

45 ORLEANS
A.M.C.
Matériaux et fournitures de bureau
15 rue des Maurier (38) 62 62 56
CHALETTE SUR LOING
Libraire-Papeterie des Ecoles
46 bis rue Mirabeau (38) 05 43 71
MONTARGIS
MAJUSCULE
3 rue de Loing (38) 85 52 81

47 AGEN
S.A.R.L. DALI
Librairie des Écoles THOMAS
16 rue Régie Condorcet (53) 66 13 02

49 ANGERS
J.-P. ADRIEN - S.E.L.P.E.L.
28 rue St Julien (41) 88 07 16
M.B. BROUILLET
30 bd Carnot (40) 88 35 95
CHOLET
M.B. BROUILLET
1 rue St Pierre (41) 62 53 74
SAUMUR
M.B. BROUILLET
77 rue d'Orléans (40) 51 35 25

50 TOURLAVILLE (CHERBOURG)
HAFERT
Équipement et fournitures de Bureau
Rue des Mêlées –
Z.A. bd de l'Est (33) 93 32 90

51 REIMS
QUERLIN MARTIN ET FILS
82 place Drouet d'Erlon (26) 88 40 30

52 JOINVILLE
BOLOT DEOFFRIN
53, rue Aristide Briand
B.P. 20 (26) 94 14 63

54 LUNEVILLE
BASTIEN
23 à 29, rue Clemenceau (8) 372 31 21

56 LORIENT
J. CRAFF
1 place Aristide Briand (97) 64 32 57
PONTIVY
GILLES BLAYO
30 rue Nationale B.P. 107 (37) 25 04 92

57 METZ
PAUL EVEN
1 rue Ambroise-Thomas (8) 775 67 11
SARREGUEMINES
PIERRON MULLER
5 rue Sainte-Croix B.P. 365 (8) 709 07 01

58 COSNE SUR LOIRE
D. ROBIN
6 rue du Commerce (86) 26.06.18

59 LILLE
S.E.G. DAUPHINOR
35 rue Faidherbe (20) 54 87 16
DENAIN
LOZE BERTRAND
106 rue de Villars (27) 44 06 88
DUNKERQUE
MAJUSCULE
Place Jean Bart (28) 65 26 63
Librairie Papeterie Universitaire DEMEY
1 rue du Lycée (28) 66 96 54
HAZEBROUCK
BURO SLANDRES
6 bis Avenue Lemire (28) 43 36 02
TOURCOING
LAMARTINE
30 place de la République (20) 24 36 21

60 BEAUVAIS
QUENEUTTE
5 rue du Dr Gérard (4) 445 12 74 +

CREIL
QUENEUTTE
22 rue de la République (4) 425 04 26 +

61 LA FERTE MACE
BERNARD GALLOT
Maison de la Presse
Place du Général Leclerc (33) 37 22 43

62 MONTAGNE AU PERCHE
GAUTIER S.A.R.L.
48 place Charles-de-Gaulle (33) 25 01 17

62 ARRAS
BRUNET
21 rue Gambetta (21) 23 46 34
CALAIS
BUROCONFOR
67 bd Fleurus (21) 96 30 10
BOULOGNE SUR MER
DUMINY
Angle rue Lafayette et rue Victor Hugo (21) 30 09 25

63 CLERMONT FERRAND
LES VOLCANS D'AUVERGNE
80 bis de Clermont (73) 33 04 40

64 PAU
TONNET
Place Marguerite-Labarde (59) 30 77 33
BAYONNE
JACQUES MURILLO
13 place Fréteau (59) 59 82 08

65 TARBES
BIGORRE BUREAU
14 rue Abbé Torne
(62) 34 73 94 (62) 34 07 33

66 PERPIGNAN
BRUN FRERES
22 rue Ste Augustine (68) 34 41 53

69 VILLEFRANCHE SUR SAONE
DEVELAY
Librairie des Écoles
105 rue Nationale (74) 65 01 49
50 rue Nationale (78) 65 08 36

71 MACON
RENAUDIER S.A.R.L.
21 quai Lamartine (85) 38 32 96
LE CREUSOT
RENAUDIER
72 rue Lecherc (85) 55 00 42

72 LE MANS
Galerie du Livre DOUCET
81 av. du Général-de-Gaulle (43) 24 49 20

76 THONON LES BAINS
RIFMANN
7 rue des Arts (50) 71 25 45

75 PARIS 16e
LAMARTINE
101 rue de la Pompe (1) 727 31 31

76 ROUEN
LE BUREAU MODERNE
Mme Jean DELAMARE
24 rue St Nicolas (35) 71 48 02
YVETOT
Maison DELAMARE
37 bis Mail - 11 rue des Victoires
(32) 95 02 58

77 MELUN
JACQUES AMYOT
22 rue Paul Doumer (1) 452 03 81 +

78 VERSAILLES
RUAT
30 avenue de St-Cloud
(2) 950 21 41 (3) 953 94 53
MANTES LA JOLIE
TONNERX
47 rue Nationale
4-6, rue du Collège (3) 943 32 37

CHATOU
SOFIP
Bouchot 1 rue Paul Abeille (3) 952 73 37
Ecouen
10 bis, rue Marcelin Berthelot
78 GRESSURE
M.B. BROUILLET
2 place Carnot (46) 97 13 43

80 ABBEVILLE
DUCLERC
33 place de l'Hôtel de Ville
BUROMODERN
20 place de la Lanterne (22) 24 21 92

81 ALBI
VERDEILLE
Librairie des Écoles
62 rue Croix Verte (63) 54 07 43

83 TOULON
BONNAUD
7 Théâtre 3 de Strasbourg (94) 22 27 62

84 AVIGNON
AMBLARD
13-14 portail Matheron (90) 82 51 21
ART
DUMAS
91 rue des Marchands (90) 74 23 31

85 LA ROCHE SUR YON
CHAGNEAU
1 rue de la Vieille Horloge (51) 37 30 58

86 POITIERS
A. GRANVAUD
38 rue Gambetta (46) 41 15 53

88 EPINAL
HOMEYER
2 rue du 170e R.I. (29) 54 01 20

89 AUXERRE
DUFLOUX
11 13-15 rue de la Draperie (85) 52 00 12

91 ARPAJON
LEMONNIER
3 rue Dourdan (6) 488 103 96

92 VITRY SUR SEINE
LENFANT
2 avenue du Parc (1) 680 05 48

95 PONTOISE
C.A.P.
J. Pierre SLIAH
Rue Angle n° 21
La Bureautique
13 rue Pothier (3) 036 26 30

Aujourd'hui aucun domaine n'est plus évolutif que celui des micro-ordinateurs familiaux. Les modèles se suivent à un rythme élevé, apportant sans cesse de nouvelles améliorations voire des innovations déterminantes. Pour vous guider dans votre choix et vous aider à vous procurer l'appareil le mieux adapté à l'usage que vous lui destinez, faites confiance aux spécialistes MAJUSCULE de votre région.
MAJUSCULE, c'est le seul réseau aussi étendu de micro-ordinateurs familiaux et de consommables informatiques.
L'importance de ce groupe, plus de cent distributeurs en France, lui permet de vous proposer un très large choix de grandes marques. Vous y trouverez l'accueil, le conseil et le séneux des spécialistes. Choisissez MAJUSCULE près de chez vous.

MAJUSCULE, le réseau conseil des produits de pointe

Un spécialiste •MAJUSCULE• vous attendrès de chez vous

Demandez notre catalogue consommables informatiques.

BEAC

Référence 105 du service-lecteurs (page 172)

PSI : LES LIVRES DE LA DÉCOUVERTE

SHARP

La découverte du PC-1251
par Jean Pierre Richard
Comment aborder la programmation du Sharp PC-1251 ? Après une étude fonctionnelle de la machine la programmation en langage Basic est étudiée de façon progressive. Citons parmi les thèmes traités instructions et commandes variables et mémoires sous programmes et traitements de chaines gestion des peripheriques Quelques exercices d application et un index viennent compléter ce manuel qui fournira à l utilisateur tous les elements de base indispensables à la programmation en basic
224 pages - 92,00 FF / 710,00 FB

La découverte du PC-1500
par Jean-Pierre Richard
Un ouvrage pour les néophytes curieux qui veulent en savoir plus sur leur ordinateur de poche PC-1500 (ou TRS 80 PC 2) Quelles instructions et commandes emploie-t-il ? Quels types de variables utilise t-il ? Comment la mémoire est-elle structurée ? Toutes ces questions et bien d autres trouvent leur reponse dans ce livre Un manuel complété d exercices d exemples d application et d un index.
208 pages - 92,00 FF/710,00 FB

CASIO

La découverte du FX-702 P
par Jean-Pierre Richard
Instructions et commandes variables et mémoires fonctions périphériques cet ouvrage fournit aux débutants tous les éléments de base nécessaires à la programmation en langage Basic de la Casio FX 702 P Il est complété de nombreux exemples et exercices d application
216 pages - 82,00 FF/710,00 FB

La découverte du PB-100
par Patrick Morgoccu
Ce livre dévoile progressivement toutes les recettes de l ordinateur de poche Casio PB 100 et du Basic à l aide de nombreux exemples d application permettant aussi aux novices d entrer en douceur dans le monde de l informatique
160 pages - 82,00 FF/635,00 FB

APPLE

La découverte de l'Apple II
Apple IIe et Apple II plus
par D Schmen et F Lévy
Apple IIe ou Apple II plus ? Peu importe Ce livre bien qu orienté sur Apple IIe vous initiera en douceur au merveilleux de ces deux générations d Apple et à leur langage de programmation le Basic Applesoft. Sa démarche progressive et ponctuée d exemples en fait un véritable ouvrage pour débutants Ceux ci pourront tester grâce aux exercices en fin de chapitre la bonne assimilation des concepts étudiés
82,00 FF/635,00 FB

TEXAS INSTRUMENTS

La découverte de la TI-57
par Xavier de la Tullaye
S adressant aux débutants cet ouvrage les conduit dans un langage clair de l élémentaire 2 + 2 à des programmes perfectionnés Après une étude fonctionnelle de la calculette la programmation est expliquée progressivement de la conception à la réalisation en s appuyant sur de nombreux exemples
144 pages - 72,00 FF/555,00 FB

La découverte de la TI-99/4 A
par Frédéric Levy
Cet ouvrage d initiation conçu pour les utilisateurs de TI 99/4 A desireux d apprendre la programmation en Basic s articule selon trois points principaux les instructions classiques du Basic telles qu elles existent sur le TI 99/4 A ses possibilités graphiques (dessins et animation) ses ressources musicales
144 pages - 82,00 FF/635,00 FB

COMMODORE

La découverte du VIC
par Daniel-Jean David
Ce livre qui comprend essentiellement une presentation progressive du langage Basic fait découvrir les divers champs d application de l ordinateur individuel VIC Il aborde spécialement les points forts de ce PSI graphiques sons couleurs
176 pages - 82,00 FF/655,00 FB

La découverte du CBM
par Daniel Jean David
Cet ouvrage d initiation à l utilisation de la gamme PET/CBM comprend essentiellement une introduction progressive au langage Basic La découverte du langage est conduite en bâtissant des programmes de plus en plus élaborés au fur et à mesure de l introduction de notions nouvelles
136 pages - 72,00 FF / 555,00 FB

La découverte du Commodore 64
par Daniel-Jean David
Le C64 est un ordinateur individuel qui permet à la fois des applications sérieuses et les jeux Ce livre d initiation couvre les deux aspects Il ne nécessite pas de connaissances préalables La découverte du langage Basic est conduite en bâtissant des programmes par améliorations successives au cours desquelles les notions nouvelles s introduisent tout naturellement On aborde spécialement les points forts du C64 graphiques sons couleurs haute résolution et sprites
176 pages - 82,00 FF / 635,00 FB

ET AUSSI...

La découverte du Dragon
par Frédéric Levy et Dominique Schraen
Cet ouvrage s adresse aux utilisateurs du Dragon 32 souhaitant s initier à la programmation en Basic Conçu pour le débutant son approche est progressive attrayante et agrémentée de nombreux exemples et exercices Après un apprentissage des instructions classiques du Basic le lecteur découvre le jeu particulièrement puissant des instructions graphiques Il peut alors se lancer dans la création de ses propres programmes
144 pages - 82,00 FF/635,00 FB

Basic pour tous
par J Boisgontier et S Brabion
Apprenez l ordinateur ce livre en main placez vous devant votre ordinateur (Microsoft 5 TRS 80 Apple TO7) et commencez à écrire quelques instructions Très vite vous assimilerez les notions fondamentales de la programmation variables tests boucles Vous aurez alors compris l ordinateur et ce que vous pouvez en attendre Mais ce n est pas tout Grâce aux exemples illustrés et aux programmes commentés vous acquérirez sans difficulté les bases nécessaires qui vous permettront d écrire vos propres programmes (gestion d'enseignement jeux)
160 pages - 82,00 FF / 710,00 FB

Oric-1 pour tous
par J Boisgontier et S Brabion
Apprenez l Oric 1 1 Oric 1 pour tous ce livre en main placez vous devant votre micro ordinateur et commencez à écrire quelques instructions Vous assimilerez très vite les notions fondamentales de la programmation variables tests boucles Vous pourrez alors aborder la magie de l Oric les **graphiques** et les **sons**, grâce aux nombreux exemples illustrés et aux programmes directement commentés Sur ces bases il vous sera aisé d approfondir vos connaissances et d écrire vos propres programmes (gestion d'enseignement jeux)
176 pages - 82,00 FF / 710,00 FB

La découverte de l'Oric
par Daniel Jean David
Après une introduction formée de rappels généraux sur l informatique cet ouvrage comprend essentiellement une presentation progressive du langage Basic La découverte du langage est conduite en bâtissant des programmes par améliorations successives au cours desquelles les notions nouvelles s introduisent naturellement On aborde spécialement les points forts de l Oric graphiques sons couleurs horloge
176 pages - 82,00 FF/635,00 FB

La découverte de TO7
par D Schmen et M Charbit
Cet ouvrage s adresse aux utilisateurs du TO 7 souhaitant s initier à la programmation en Basic Conçu pour le débutant, son approche est progressive attrayante et agrémentée de nombreux exemples et exercices Après un apprentissage des instructions classiques du Basic, le lecteur découvre l attrait des instructions graphiques ainsi que l utilisation du crayon optique Il peut alors se lancer dans la programmation de ses propres jeux vidéo
176 pages - 82,00 FF/635,00 FB

AGRAPH

DESIGNATION	PRIX

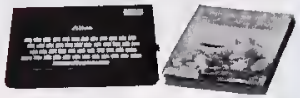
PROMOTION SPECIAL NOEL

LA BOUTIQUE INFORMATIQUE POUR L'ENTREPRISE

PROMOTION

APPLE IIe 64 K - Clavier Azerty
+ moniteur Philips + floppy

ALICE



Micro-ordinateur d'initiation. Langage BASIC. 9 couleurs. Mémoire 8 Ko ROM. Connectable sur tout téléviseur. Avec livre-guide.
1.195 F t.t.c.

2990 F

LYNX 48 K

Sinclair ZX 81
580 F

ORIC I

~~5050 F~~
4100 F

ORIC 48 K + manuel fr.
+ Cordon Peritel
+ Imprimante 4 couleurs

NOUVELLE ADRESSE

67, cours Lieutaud 13006 MARSEILLE
Boutique : 42.99.42 *S.A.V.* : 78.92.75

Boutique à Toulon
270, Bd Foch - Immeuble Le Concorde
Tel 94 / 93.11.20
TOULON

Référence 104 du service-lecteurs (page 172)

Complétez votre information grâce au service lecteurs en utilisant la carte ci-contre

(Cerclez les numéros des différentes informations qui vous intéressent)

BULLETIN DE COMMANDE

Pour vous abonner
(pour commander des numéros)
à

le magazine de l'informatique pour tous

utilisez cette carte
(Voir au verso)

SERVICE PETITES ANNONCES
39 rue de la Grange-aux-Belles
75484 Paris Cedex 10
FRANCE

L'ORDINATEUR INDIVIDUEL - N° 54 bis DEC. 1983
Ne pas utiliser cette carte plus d'un an après sa parution

SERVICE LECTEURS

Votre nom _____
Adresse _____
Profession _____
Etes-vous abonné ? ☐ oui ☐ non

Critiques, suggestions, souhaits... que nous lirons avec la plus grande attention et que nous publierons éventuellement

REDACTION

PUBLICITE

L'ORDINATEUR INDIVIDUEL (service abonnement)
39 rue de la Grange-aux-Belles, 75484 Paris Cedex 10

☐ Je souscris un abonnement pour un an (11 numéros) à L'ORDINATEUR INDIVIDUEL (Tarif France : 198 FF TVA 4 % incluse, Belgique : 1 650 FB + 100 FB de port, Etranger ou Etudiant : voir ci-contre).

☐ *Je désire recevoir les numéros antérieurs suivants. (Prix d'un numéro France : 22 FF, Etranger : voir ci-contre.)* _____

par : chèque postal ☐ chèque bancaire ☐ d'un montant de _____

M/Mme/Mlle _____
Profession _____
Adresse _____
_____
Pays _____ Code postal ☐☐☐☐☐ Ville _____

La photocopie de ce bulletin rempli constitue une pièce justificative légale du règlement effectué. Aucune facture ne sera établie par nos services.

Cadre réservé à nos services ☐☐☐ ☐☐☐ ☐☐☐ ☐ ☐☐☐ ☐☐
214 223

54 bis

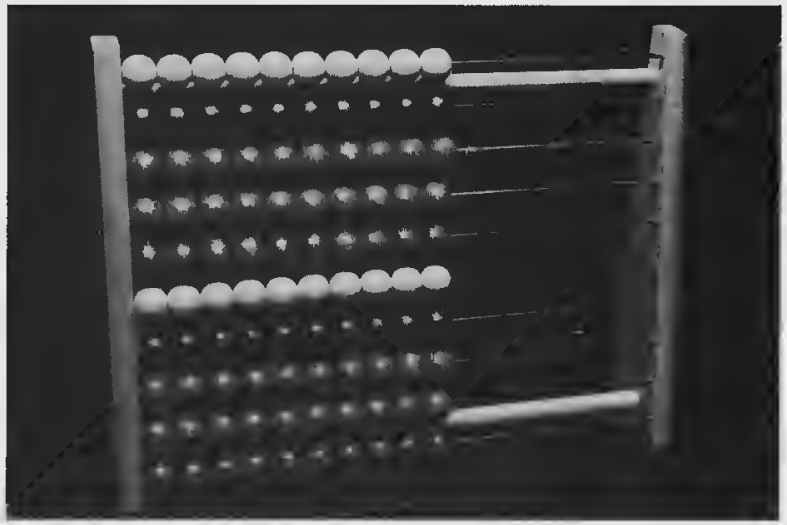
PETITES ANNONCES GRATUITES *
de L'ORDINATEUR INDIVIDUEL

Complétez la grille ci-dessous en lettres d'imprimerie en utilisant une division par lettre/signe ou espace. En aucun cas le message ne doit dépasser les cinq lignes de 36 caractères, adresse comprise. Attention, seule l'adresse complète est admise, ni boîte postale, ni téléphone. Pour les ventes de matériels d'occasion, indiquez le mois et l'année d'achat au fournisseur.

Cochez la case (une seule) de la rubrique où vous désirez voir figurer votre PA
☐Clubs ☐Contacts ☐Recherche de programmes ☐Recherche de matériels ☐Vente de matériels ☐Divers

Matériel concerné (le cas échéant)
☐Ordinateur individuel de table ☐Ordinateur de poche ☐Divers Êtes-vous abonné ? ☐ oui ☐ non

☐☐☐☐☐



Si vous savez vous servir de cet instrument, vous saurez sûrement faire tourner nos logiciels en deux heures.

Avec un peu de bonne volonté, bien sûr.

Les logiciels Saari ne sont pas faits pour des programmateurs émérites ou des informaticiens avertis.

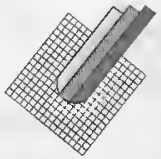
Les logiciels Saari sont suffisamment faciles à utiliser pour que vous, Directeur d'une PME, Chef Comptable, Avocat, Expert comptable, etc., tiriez le meilleur parti de la Comptabilité Saari, de la Paie Gipsi, de la Gestion de Dossiers ou du Programme de Facturation et Stock.

Les manuels sont bien clairs, écrits en bon français et illustrés. Et, si vous avez un doute sur une manœuvre, une touche ou une opération, n'hésitez pas à utiliser le Saari Sécurité Service.

S'il vous fallait quelques preuves des qualités des logiciels Saari, sachez qu'ils sont contrôlés par le Bureau Véritas et que la Paie Gipsi a obtenu la Pomme d'Or, récompense suprême décernée par Apple.

La presse Informatique a, elle aussi, beaucoup parlé des logiciels Saari, décernant ses meilleures notes à leur facilité d'apprentissage ou d'utilisation, ainsi qu'à leur richesse fonctionnelle, leur sécurité d'emploi ou leur finition.

Les quatre logiciels Saari vous rendront de gros services, sans jamais vous apporter de gros soucis.



saari®

« L'Esprit Français »

45 rue Galilée 75116 Paris
Tél (1) 723 78 56 Télex 611 423

DAGNE AUFFARIS

Pour toutes précisions sur la société ou le produit présenté ci-dessus : Référence 115 du service-lecteurs (page 172)

PARLEZ-MOI BASIC

Jean-Pierre Brunerie
Ludwick Zuber

Vous vous êtes peut-être souvent arraché les cheveux devant le programme Basic d'un autre ordinateur, et que vous aimeriez bien adapter sur le vôtre. Que vous soyez chevronné ou débutant, le tableau que nous vous présentons vous sera d'une aide efficace.

Basic signifie « de base », « ce qui est le plus simple ». Basic est aussi le langage qu'utilisent aujourd'hui six millions d'ordinateurs individuels de par le monde. Le langage ? Oui, dans un certain sens, mais, malheureusement, suivant l'ordinateur que vous avez, son Basic présentera un ou des traits spécifiques et il convient donc de parler des langages Basic. Pour vous aider à adapter les programmes Basic, nous avons établi un tableau comparatif qui, pour environ cent instructions en Basic Microsoft, donne leur « traduction » sur dix-huit ordinateurs individuels courants.

L'adaptation de programmes Basic d'un OI à un autre ne pose, dans 90 % des lignes, que peu ou pas de problèmes. RUN, NEW, END, RETURN, etc., sont pratiquement toujours les mêmes.

Oui, mais que se passe-t-il avec les 10 % restant ? Ces 10 % renferment toutes les variantes, tous les faux amis, toutes les instructions qui changent de sens, toutes les possibilités d'astuces bien particulières (indubitablement les plus prisées...), bref tous les « patois » Basic imaginables.

Prenons par exemple un Vic 20 et un Apple : en rencontrant l'instruction INPUT de lecture sur disque de l'Apple, vous pourriez penser qu'il s'agit de la saisie au clavier du Vic 20, et modifier le programme en conséquence. L'exemple est simple mais GET, RND, CLR et autres CLS reservent, hélas, des surprises autrement désagréables.

Attention aux instructions qui s'adressent à la mémoire de votre ordinateur. Les ordres PEEK, POKE, CALL et autres USR devront pratiquement systématiquement être adaptés. Les symboles graphiques, la couleur et le son sont eux aussi très spécifiques à l'ordinateur utilisé et, là encore, il est préférable de recréer plutôt que de traduire d'un ordinateur à un autre.

Pour le reste, vous allez retrouver dans les pages qui suivent environ cent instructions Basic Microsoft classées par ordre alphabétique.

Ainsi, ce tableau pourra-t-il être utilisé, non seulement comme dictionnaire des instructions Basic, mais aussi comme table de traduction des différents ordres d'un matériel à un autre.

Il est à noter que, pour des raisons évidentes d'encombrement (compte-tenu du nombre important d'instructions, beaucoup étant spécifiques à un matériel), ce tableau ne saurait être exhaustif. Comme il est plus descriptif qu'explicatif, il conviendra de l'utiliser en parallèle avec le manuel d'utilisation des matériels considérés.

Table décroisée

Certaines adresses étant spécifiques à une machine, ce tableau vous permet de retrouver, par ordinateur, à quel ordre Basic Microsoft elles correspondent. Il complète donc celui présenté dans les pages suivantes.

Apple		
DEL	→	DELETE
GET	→	INKEYS
RENAME	→	NAME
NOTRACE	→	TROFF
TRACE	→	TRON
Ateri 400		
CLR	→	CLEAR
CLOG	→	CLG
BYE	→	SYSTEM
GRAPHICS	→	TEXT
Atom		
CH	→	ASC
LINK	→	CALL, USR
$	→	CHR$
BBC		
LOG	→	CLG
LN	→	LOG
BPUT	→	OUT
RENUMBER	→	RENUM
TRACE OFF	→	TROFF
TRACE ON	→	TRON
Commodore 64		
CLR	→	CLEAR
Osi		
ACOS	→	ACS
ASIN	→	ASN
LOGT	→	CLG
DOT	→	PLOT
Dragon 32		
EXEL	→	CALL
DEL	→	DELETE
MEM	→	FRE
PSET	→	PLOT
Lynx		
ARCOS	→	ACS
ARCSIN	→	ASN
ARCTAN	→	ATN
LOG	→	CLG
DEL	→	DELETE
ROUND	→	FIX
GET$ KEYS	→	INKEY$
LN	→	LOG
TRACE OFF	→	TROFF
TRACE ON	→	TRON
MZ 80 A		
CLR	→	CLEAR
MEM	→	FRE
GET	→	INKEY$
LN	→	LOG
BYE	→	SYSTEM
Oric 1		
GET, KEY$	→	INKEY$
LN	→	LOG
CURSET	→	PLOT
TI 99		
NUM	→	AUTO
OLD	→	CLOAD
CALL CLEAR	→	CLS
CON	→	CONT
RES	→	RENUM
UNTRACE	→	TROFF
TRACE	→	TRON
TRS 80 modèle 1		
CINT	→	INT
SET	→	PLOT
SPACE$	→	SPC
Vic 20		
SYS	→	CALL
CLR	→	CLEAR
GET	→	INKEY$
ZX 81		
LN	→	LOG
ZX Spectrum		
LN	→	LOG

Première parution L'OI n° 53

Instructions Basic (ABS à CHR$)

Ordre Basic (Basicois) / Ordinateur individuel	ABS (ABS) Valeur absolue	ACS (ACS) Arc cosinus	AND (ET) Opérateur logique signifie ET	ASC (ASC) Donne le code ASCII du premier caractère de la chaîne	ASN (ASN) Arc sinus	ATN (ATN) Arc tangente	AUTO (AUTO) Numérotation automatique des lignes de programme	CALL (APPELLE) Appel d'une routine en langage machine	CHAIN (ENCHAINE) Appel d'un nouveau programme	CHR$ (CARS) Donne le caractère dont le code ASCII est arg
Alice (Tandy MC 10)	ABS (arg)		AND	ASC (chaî)						CHR$ (arg)
Apple	ABS (arg)		AND	ASC (chaî)		ATN (arg)		CALL adr	CHAIN « nom programme »	CHR$ (arg)
Atari 400	ABS (arg)		AND	ASC (chaî)		ATN (arg)			RUN « C: »	CHR$ (arg)
Atom	ABS (arg)	ACS (arg)	AND	CH (chaî) CH « »	ASN (arg)	ATN (arg)	OPTION	LINK label		$ (arg)
BBC	ABS (arg)	ACS (arg)	AND	ASC (chaî)	ASN (arg)	ATN (arg)	AUTO [n° ligne début pas]	CALL adr [var]	CHAIN « nom programme »	CHR$ (arg)
Commodore 64	ABS (arg)		AND	ASC (chaî)		ATN (arg)		SYS (adr)		CHR$ (arg)
Dai	ABS (arg)	ACOS (arg)	IAND	ASC (chaî)	ASIN (arg)	ATN (arg)		CALLM adr [arg]		CHR$ (arg)
Dragon 32	ABS (arg)		AND	ASC (chaî)		ATN (arg)		EXEC adr		CHR$ (arg)
Hector (ex Victor)	ABS (arg)	ASC (chaîn [exp])	AND	ASC (chaî)		ATN (arg)	AUTO [n° ligne début] [pas]	USR adr [mémoire] [param]		CHR$ (arg)
Lynx	B (arg)	ARCCOS (arg)	AND	ASC	ARCSIN (arg)	ARCTAN (arg)	AUTO [n° ligne début [pas]]	CALL		CHR$ (arg)
MZ 80 A	ABS (arg)		AND	ASC (chaî)		ATN (arg)				CHR$ (arg)
Oric 1	ABS (arg)		AND	ASC (chaî)		ATN (arg)		CALL adr		CHR$ (arg)
PET/CBM	ABS (arg)		AND	ASC (chaî)		ATN (arg)	AUTO pas	SYS (adr)	SWAP « nom prog »	CHR$ (arg)
Thomson TO7	ABS (arg)		AND	ASC (chaî)			AUTO [n° ligne début [pas]]		RUN MERGE « nom prog »	CHR$ (arg)
TI 99	ABS (arg)		AND	ASC (chaî)		ATN (arg)	NUM [n° ligne début] [pas]]			CHR$ (arg)
TRS 80 modèles I et 3	ABS (arg)		AND	ASC (chaî)		ATN (arg)	AUTO [n° ligne début pas]		CHAIN « nom programme »	CHR$ (arg)
Vic 20	ABS (arg)		AND	ASC (chaî)		ATN (arg)		SYS adr		CHR$ (arg)
ZX 81	ABS (arg)	ACS (arg)	AND	CODE (chaî) N.B. ZX 81 n'utilise pas ASCII	ASN (arg)	ATN (arg)		LET var = USR (adr)		CHR$ (code) N.B. code ≠ ASCII
ZX Spectrum	ABS (arg)	ACS (arg)	AND	CODE (chaî) N.B. Spectrum n'utilise pas ASCII	ASN (arg)	ATN (arg)		LET var = USR (adr) (possibilités multiples)		CHR$ (code) N.B. code ≠ ASCII

arg : argument numérique **exp** : expression **chaî(n)** : chaîne numérique **adr** : adresse **var** : variable (nom de variable) [ ] : indique une clause optionnelle ( ) : indique des clauses équivalentes

Instructions Basic (CLEAR à CSAVE)

Ordre Basic (Basicois) / Ordinateur individuel	CLEAR (RAZ) Remise à zéro	CLG (LOGD) Logarithme en base 10	CLOAD (CCHARGE) Chargement en mémoire d'un fichier sur cassette	CLOSE (FERME) Fermeture de fichier	CLS (EFF) Effacement d'écran	CONT (CONTINUE) Relance une exécution stoppée	COLOR (COULEUR) Définition de couleur	COS (COS) Cosinus	COUNT (COMPTE) Compte les caractères imprimés depuis RETURN	CSAVE (CCOPIE) Sauvegarde d'un fichier sur cassette
Alice (Tandy MC 10)	CLEAR (arg)		CLOAD nom prog & CLOAD x, fich		CLS	CONT	CLS (arg) (efface l'écran)	COS (exp)		CSAVE « nom » CSAVE x fich.
Apple	CLEAR		LOAD [nom fich] (idem disquette)	CLOSE « nom »		CONT	COLOR = arg	COS (exp)		
Atari 400	CLR	CLOG (arg)	CLOAD « nom fich »	CLOSE [#n° fich]		CONT	COLOR arg	COS (exp)		CSAVE [« nom fich »]
Atom			LOAD [ ]				COLOR arg	COS (exp)	COUNT	SAVE « »
BBC	CLEAR	LOG (arg) N.B. CLG a une autre signification (→ cf CLS)	LOAD « nom fich » (idem disquette)	CLOSE # n° fich N.B. CLOSE # ≠ ferme tous les fichiers	CLS (écran texte) CLG (écran graph) (cf HOME)		COLOUR arg (GCOL en graphique)	COS (exp)	COUNT	SAVE [« nom fich »]
Commodore 64	CLR		LOAD [« nom fich »] (idem disquette)	CLOSE n° fich		CONT		COS (exp)		SAVE [« nom fich »] (idem disquette)
Dai	CLEAR (arg)	LOGT (exp)	LOAD [« nom fich »] (idem disquette)			CONT	COLORT arg 1 2 3 4 : text COLORG arg 1 2 3 4 : graph	COS (exp)		SAVE (mémoire) (cassette) (idem disquette)
Dragon 32	CLEAR (arg)		CLOAD [« nom fich »]	CLOSE n° fich	autre signification (cf COLOR)	CONT	CLS arg (COLOR est aussi utilisé)	COS (exp)		CSAVE « nom fich »
Hector (ex Victor)	CLEAR [exp] [adr]		LOAD [« nom fich »]		CLS [exp]	CONT	COLOR exp exp exp exp	COS (exp)		CSAVE [« nom fich »]
Lynx		LOG (exp)	LOAD « nom fich » (idem disquette)		VDU 4	CONT	INK arg	COS (exp)		SAVE « nom fich » [n° de la 1re ligne] (idem disquette)
MZ 80 A	CLR	LOG (exp)	LOAD/T « nom fich »	CLOSE [# n° fich]		CONT		COS (exp)		SAVE/T « nom fich »
Oric 1	CLEAR		CLOAD « [nom fich] » [S] N.B. S : 300 baude			CONT		COS (exp)		CSAVE « nom fich » [S] [AUTO]
PET/CBM	CLR		LOAD [« nom fich »] (idem disquette)	CLOSE # n° fich		CONT		COS (exp)		SAVE [« nom fich »]
Thomson TO7	autre utilisation		LOAD « nom fich » (idem disquette)	CLOSE	CLS	CONT	COLOR arg 1 arg 2 (utilisation peut être manuel)	COS (exp)		SAVE « nom fich » (idem disquette)
TI 99			OLD nom fich		CALL CLEAR	CON		COS (exp)		SAVE nom fich
TRS 80 modèles I et 3	CLEAR [(arg)]		CLOAD « [nom fich] »	CLOSE # n° fich [n° fich]	CLS	CONT		COS (exp)		CSAVE « nom fich »
Vic 20	CLR		LOAD [« nom fich »]	CLOSE # n° fich		CONT		COS (exp)		SAVE [« nom fich » code contrôle]
ZX 81	CLEAR		LOAD « [nom fich] » (idem disquette)		CLS	CONT		COS (exp)		SAVE « nom fich »
ZX Spectrum	CLEAR		LOAD « nom fich » (possibilités multiples)	CLOSE # n° fichier	CLS	CONTINUE	INK arg PAPER arg	COS (exp)		SAVE « nom fich » (possibilités multiples)

arg : argument numérique **exp** : expression **chaî(n)** : chaîne numérique **adr** : adresse **var** : variable (nom de variable) [ ] : indique une clause optionnelle ( ) : indique des clauses équivalentes

Instructions Basic (DATA à FLASH)

Ordre Basic (Basicals) / Ordinateur individuel	DATA (INFOS) Création d'une liste d'éléments	DEF FN (DEF FN) Définition de fonctions utilisateur	DELETE (ENLEVE) Effacement de lignes de programme	DIM (DIM) Définition de tableaux	EDIT (EDIT) Edite une ligne de programme	END (FIN) Fin d'exécution d'une séquence	EOF (EOF) Fin de fichier	EXP (EXP) Exponentielle	FIX (FIX) Valeur arrondie	FLASH (CLIGNE) Clignotement
Alice (Tandy MC 10)	DATA var [var]			DIM nom (arg)		END		EXP (exp)		
Apple	DATA var [var]	DEF FN var (var) = exp	DEL n° ligne 1 [n° ligne 2]	DIM nom (arg.)	touche ESC + touche CTRL	END		EXP (exp)		FLASH
Atari 400	DATA var [var]			DIM nom (arg)		END		EXP (exp)		
Atom	OPTION			DIM non équivalent		END		EXP (exp)		
BBC	DATA var [var]	DEF FN var [(var var)] = exp	DELETE n° ligne 1 n° ligne 2	DIM nom (arg.)	sélection par curseur et touche COPY	END	EOF #	EXP (exp)		VDU
Commodore 64	DATA var [var]	DEF FN var = exp		DIM nom (arg)		END		EXP (exp)		
Dai	DATA var [var]			DIM nom (arg)		END		EXP (exp)		
Dragon 32	DATA var [var]	DEF FN var (var) = exp	DEL n° ligne 1 n° ligne 2	DIM nom (arg)	EDIT n° ligne	END	EOF (-1)	EXP (exp)	FIX (arg)	
Hector (ex Victor)	DATA var [var]	DEF FN var (var) = exp		DIM nom (exp)	EDIT n° ligne EDIT (passe en mode édition)	END		EXP (exp)		FLASH n° couleur [, exp]
Lynx	DATA var [var]		DEL n° ligne 1 [n° ligne 2]	DIM nom (arg)	utilisation touche CTRL	END		EXP (exp)	ROUND ON/OFF	
MZ 80 A	DATA var [var]		DELETE (FD n° disque « nom fich » (efface tout fichier)	DIM nom (arg)		END	EOF (# n° de fichier)	EXP (exp)		
Oric 1	DATA var [var]	DEF FN var (var) = exp		DIM nom (arg)	EDIT n° ligne	END		EXP (exp)		
PET/CBM	DATA var [var]	DEF FN var = exp	DELETE n° ligne 1 n° ligne 2	DIM nom (arg)		END		EXP (exp)		
Thomson TO7	DATA var [var]		DEEETE n° ligne 1 [(n° ligne 2)]	DIM nom (arg.)		END	EOF	EXP (exp)	FIX (arg)	
TI 99	DATA var [var]	DEF var (var) = exp			EDIT n° ligne	END	EOF	EXP (exp)		
TRS 80 modèles 1 et 3	DATA var [var]	différentes formulations possibles	DELETE n° ligne 1 n° ligne 2	DIM nom (arg.)	EDIT n° ligne	END		EXP (exp)	FIX (arg)	
Vic 20	DATA var [var]	DEF FN (var) = exp	DELETE [n° ligne 1) [ ] (n° ligne 2]	DIM nom (arg)		END		EXP (exp)		
ZX 81				DIM nom (arg.)	(sélection par curseur)			EXP (exp)		
ZX Spectrum	DATA var [var]	DEF FN var [(var, var)] = exp		DIM nom (arg.)	EDIT [n° ligne] (sélection par curseur)			EXP (exp)		PRINT FLASH 1

arg. : argument numérique ; **exp.** : expression ; **chai(n)** : chaîne numérique ; **adr** : adresse ; **var** : variable (nom de variable) ; **[ ]** : indique une clause optionnelle ; **|** : indique des clauses équivalentes.

Instructions Basic (FOR à INPUT#)

Ordre Basic (Basicals) / Ordinateur individuel	FOR (REPETE) Utilisé avec NEXT pour répéter une séquence	FRE (LIBRE) Donne la place disponible en mémoire	GET (PRENDS) Lecture d'un enregistrement	GOSUB (VAVIENS) Appel d'un sous-programme	GOTO (VATEN) Branchement à une autre partie du programme	HOME (VIDE) Vide l'écran et positionne le curseur en haut à gauche	IF THEN ELSE (SI/ALORS/SINON) Si alors sinon	INKEY$ (CAPTE$) Entrée directe d'un caractère tapé au clavier	INPUT (DEMANDE) Lecture de données entrées au clavier	INPUT# (DEMANDE#) Lecture d'un fichier séquentiel
Alice (Tandy MC 10)	FOR TO [STEP]	MEM		GOSUB n° ligne	GOTO n° ligne	HOME	IF exp THEN ordres [ELSE ordres]	INKEY$	INPUT [chai n.] var	
Apple	FOR var = arg 1 TO arg 2 [STEP arg 3]	FRE (0)	INPUT var [var]	GOSUB { n° ligne / var / exp }	GOTO n° ligne	HOME	IF exp THEN ordres [ELSE ordres]	GET var	INPUT [« message »] var [var]	
Atari 400	FOR var = arg 1 TO arg 2 [STEP arg 3]	FRE (exp)	GET # n° fichier var	GOSUB { n° ligne / var / exp }	GOTO { n° ligne / var / exp }		IF exp THEN ordres		INPUT [chaîne] var [var]	
Atom	FOR var = arg 1 TO arg 2 [STEP arg 3]			GOSUB { n° ligne / label / exp. }		CTRL ou Print $30 (par programme)	IF exp THEN ordres		INPUT [chan] var [var]	
BBC	FOR var = arg 1 TO arg 2 [STEP arg 3]	HIMEM-TOP (non strictement équivalent)	INPUT # n° fichier var [var]	GOSUB { n° ligne / var / exp }	GOTO { n° ligne / var / exp }	CLS mode texte CLG mode graphique	IF exp THEN ordres [ELSE ordres]	GET$ var ou INKEY$ (tps)	INPUT (chan) [ ] var [var]	INPUT # n° fichier var [var] (idem GET)
Commodore 64	FOR var = arg 1 TO arg 2 [STEP arg 3]	FRE (exp)	GET # n° fichier var [var]	GOSUB n° ligne	GOTO n° ligne	PRINT « touche home »	IF exp THEN ordres	GET var	INPUT [« message »] var [var]	INPUT # n° fichier var [var]
Dai	FOR var = arg 1 TO arg 2 [STEP arg 3]	FRE		GOSUB n° ligne	GOTO n° ligne	?CHR$ (12)	IF exp THEN ordres (autre IF exp GOTO n° ligne)	var = GETC	INPUT [« message »] var [var ...]	
Dragon 32	FOR var = arg 1 TO arg 2 [STEP arg 3]	MEM	INPUT # n° fich enreg	GOSUB n° ligne	GOTO n° ligne		IF exp THEN ordres [ELSE ordres]	INKEY$	INPUT [« message »] var [var ...]	INPUT # 1 var [var ...]
Hector (ex Victor)	FOR var = arg 1 TO arg 2 [STEP arg 3]	FREE (exp)		GOSUB { n° ligne / var / exp }	GOTO { n° ligne / var / exp }	WIPE	IF exp THEN ordres [ELSE ordres]	INKEY$ (tps)	INPUT [« message »] var [var ...]	
Lynx	FOR var = arg 1 TO arg 2 [STEP arg 3]	MEM		GOSUB n° ligne	GOTO n° ligne	CLS	IF exp THEN ordres [ELSE ordres]	GET$ (attente) KEY$ (pas d'attente)	INPUT [« message »] var [var]	
MZ 80 A	FOR var = arg 1 TO arg 2 [STEP arg 3]	MEM	INPUT T enreg.	GOSUB n° ligne	GOTO n° ligne		IF exp THEN ordres	GET var	INPUT [« message »] var [var]	INPUT # n° fichier var [var]
Oric 1	FOR var = arg 1 TO arg 2 [STEP arg 3]	FRE (exp)		GOSUB { n° ligne / var / exp }	GOTO { n° ligne / var / exp }	PRINT CHR$ (30)	IF exp THEN ordres [ELSE ordres]	KEY$	INPUT [« message »] var	
PET/CBM	FOR var = arg 1 TO arg 2 [STEP arg 3]	FRE (exp)	GET # n° fich nr var [var]	GOSUB n° ligne	GOTO n° ligne	PRINT « touche home »	IF exp THEN ordres	GET var	INPUT [« message »] var [var]	INPUT # n° fichier var [var...]
Thomson TO7	FOR var = arg 1 TO arg 2 [STEP arg 3]	FRE (0)		GOSUB n° ligne	GOTO n° ligne	CLS	IF exp THEN ordres [ELSE ordres]	var = INKEY$	INPUT [« message »] var [var]	
TI 99	FOR var = arg 1 TO arg 2 [STEP arg 3]			GOSUB n° ligne	GOTO n° ligne		IF exp THEN ordres [ELSE ordres]		INPUT [« message »] var [var]	
TRS 80 modèles 1 et 3	FOR var = arg 1 TO arg 2 [STEP arg 3]	FRE (exp)	INPUT # n° fichier enreg [enreg.]	GOSUB n° ligne	GOTO n° ligne		IF exp THEN ordres [ELSE ordres]	INKEY$	INPUT [chan] var [var...]	
Vic 20	FOR var = arg 1 TO arg 2 [STEP arg 3]	FRE (exp)	GET A$ GET # n A$	GOSUB n° ligne	GOTO n° ligne	PRINT « touche home »	IF exp THEN ordres	GET var	INPUT [chan] var [var...]	
ZX 81	FOR var = arg 1 TO arg 2 [STEP arg 3]			GOSUB { n° ligne / var / exp }	GOTO { n° ligne / var / exp }	CLS	IF exp THEN ordres	INKEY$	INPUT var	
ZX Spectrum	FOR var = arg 1 TO arg 2 [STEP arg 3]		LOAD « DATA var ( )	GOSUB { n° ligne / var / exp }	GOTO { n° ligne / var / exp }	CLS	IF exp THEN ordres	INKEY$	INPUT [chan] var	

arg. : argument numérique ; **exp.** : expression ; **chai(n)** : chaîne numérique ; **adr** : adresse ; **var.** : variable (nom de variable) ; **[ ]** : indique une clause optionnelle ; **|** : indique des clauses équivalentes.

Instructions Basic (INSTR à LOAD)

Ordre Basic (Basicois) / Ordinateur individuel	INSTR (INTXT) Recherche de la position d'une chaîne dans une autre	INT (ENT) Partie entière	INVERSE (INVERSE) Vidéo inverse	KILL (DETRUIS) Suppression d'un fichier	LEFT$ (GAUCHE$) Fournit les caractères de gauche d'une chaîne	LEN (LONG) Donne la taille d'une chaîne	LET (FAIS) Affectation	LIST (LISTE) Affiche les lignes demandées	LLIST (LISTAGE) Liste programme ou lignes à l'imprimante	LOAD (CHARGE) Charge un programme en mémoire
Alice (Tandy MC 10)		INT (exp)		DELETE « nom fich »	LEFT$ (chain longueur)	LEN (chain)	LET var = exp	LIST [n° ligne 1] [n° ligne 2]	LLIST	
Apple		INT (exp)	INVERSE		LEFT$ (chain, longueur)	LEN (chain)	[LET] var = exp	LIST [n° ligne 1 [n° ligne 2]]		LOAD [nom fich.] (idem cassette)
Atari 400		INT (exp)			A$ (1, N)	LEN (chain)	[LET] var = exp	LIST [n° ligne 1 [n° ligne 2]]	LIST « P: »	LOAD « n° fichier nom fichier » (cassette cf CLOAD)
Atom					chan + taille – 1 »	LEN (chain)	[LET] var = exp	LIST [n° ligne 1 [n° ligne 2]]		LOAD [« nom fichier »]
BBC	INSTR (chain 1 chain 2 [arg])	INT (exp)	FLOT (utilisation spécifique cf manuel)		LEFT$ (chain longueur)	LEN (chain)	[LET] var = exp	LIST [n° ligne 1 [n° ligne 2]]	CTRL B puis LIST [n° ligne 1 ligne 2]	LOAD « nom fichier » (idem cassette)
Commodore 64		INT (exp)			LEFT$ (chain longueur)	LEN (chain)	[LET] var = exp	LIST [n° ligne 1 [– n° ligne 2]]	OPEN 1, 4 CMD 1 LIST [n° ligne 1 n° ligne 2]	LOAD « nom fichier » (cassette cf CLOAD)
Dai		INT (exp)			LEFT$ (chain longueur)	LEN (chain)	[LET] var = exp	LIST [n° ligne 1] [n° ligne 2]		LOAD [« nom fich »] (idem cassette)
Dragon 32		INT (exp)			LEFT$ (chain longueur)	LEN (chain)	[LET] var = exp	LIST [n° ligne 1 [n° ligne 2]]	LLIST [n° ligne 1 n° ligne 2]	
Hector (ex Victor)	autre utilisation	INT (exp)			LEFT$ (chain longueur)	LEN (chain)	[LET] var = exp	LIST [n° ligne 1] [ ] [n° ligne 2]	LLIST [n° ligne 1] [ ] [n° ligne 2]	LOAD [« nom fich »]
Lynx		INT (exp)	VDU 16		LEFT$ (chain, longueur)	LEN (chain)	[LET] var = exp	LIST [n° ligne 1 [n° ligne 2]]	LLIST [n° ligne 1 n° ligne 2]	LOAD « nom fichier » (idem cassette)
MZ 80 A		INT (exp)			LEFT$ (chain, longueur)	LEN (chain)	[LET] var = exp	LIST [n° ligne 1 [n° ligne 2]]	LLIST [n° ligne 1 n° ligne 2]	LOAD [FD n° disque] nom fichier
Oric 1		INT (exp)			LEFT$ (chain, longueur)	LEN (chain)	[LET] var = exp	LIST [n° ligne 1 n° ligne 2]	LLIST [n° ligne 1 n° ligne 2]	
PET/CBM		INT (exp)	RVS 1	SCRATCH « nom fich »	LEFT$ (chain, longueur)	LEN (chain)	[LET] var = exp	LIST [n° ligne 1 [– n° ligne 2]]	OPEN 1, 4 CMD 1 LIST [n° ligne 1 n° ligne 2]	LOAD « nom fichier » (cassette cf CLOAD)
Thomson TO7	INSTR	INT (exp)			LEFT$ (chain, longueur)	LEN (chain)	[LET] var = exp	LIST [ ] [n° ligne 1] [n° ligne 2]]	LIST « nom prog »	LOAD « nom fich » (idem cassette)
TI 99		INT (exp)					[LET] var = exp	LIST [ ] [n° ligne 1] [n° ligne 2]]		
TRS 80 modèles 1 et 3		CINT (exp)		KILL (nom de fichier)	LEFT$ (chain longueur)	LEN (chain)	[LET] var = exp	LIST [n° ligne 1 n° ligne 2]	LLIST [n° ligne 1 n° ligne 2]	LOAD « nom de fichier » (cassette cf CLOAD)
Vic 20		INT (exp)			LEFT$ (chain, longueur)	LEN (chain)	[LET] var = exp	LIST [[n° ligne 1] [n° ligne 2]]	OPEN 3, 4 CMD 3 LIST [n° ligne 1 – n° ligne 2]	LOAD « nom de fichier » (cassette cf CLOAD)
ZX 81		INT (exp)			chan [TO fin]	LEN (chain)	LET var = exp	LIST [n° ligne]	LLIST [n° ligne] (COPY copie l'écran sur imprimante)	
ZX Spectrum		INT (exp)	INVERSE 1		chan [TO fin]	LEN (chain)	LET var = exp	LIST [n° ligne]	LLIST [n° ligne]	

arg. : argument numérique, **exp** : expression, **chai(n)** : chaîne numérique, **adr** : adresse, **var** : variable (nom de variable) [ ] : indique une clause optionnelle, { } : indique des clauses équivalentes

Instructions Basic (LOG à ON/GOSUB)

Ordre Basic (Basicois) / Ordinateur individuel	LOG (LOG) Logarithme népérien	MAX (MAX) Maxima de deux valeurs	MID$ (PARTIE$) Donne les caractères d'une chaîne à partir du rang indiqué	MIN (MIN) Minimum de deux valeurs	MOD (MOD) Modulo	NAME (NOMME) Change le nom d'un fichier	NEW (NETTOIE) Efface le programme en mémoire	NEXT (ENCORE) Fin de boucle FOR	ON ERROR (SELON ERREUR) Branchement à une séquence d'erreur	ON/GOSUB (SELON/VA VIENS) Branchement calculé au sous-prog
Alice (Tandy MC 10)	LOG (exp)		MID$ (chain longueur)				NEW	NEXT [var]		ON var GOSUB n° ligne 1 [n° ligne 2]
Apple	LOG (exp)		MID$ (chain, début [longueur]			RENAME ancien nom nouveau nom	NEW	NEXT [var]	ON ERR GOTO n° ligne	ON var GOSUB n° ligne [n° ligne 2]
Atari 400	LOG (exp)		A$ (I, J)				NEW	NEXT var	TRAP [n° ligne var exp]	ON var GOSUB n° ligne 1 [n° ligne 2 ..]
Atom	LOG (exp)						NEW	NEXT var		
BBC	LN (exp)		MID$ (chain, début [longueur])		var = arg 1 MOD arg 2	OPEN 1, 8, 15 « RO : nom 1 : nom 2 »	NEW	NEXT [var]	ON ERROR GOTO [n° ligne] OFF	ON [exp var] GOSUB n° ligne 1 [n° ligne 2]
Commodore 64	LOG (exp)		MID$ (chain, début [longueur])		var = arg 1 MOD arg 2		NEW	NEXT [var]		ON var GOSUB n° ligne 1 [n° ligne 2]
Dai	LOG (exp)		MID$ (chain, début longueur)				NEW	NEXT [var]		ON var GOSUB n° ligne 1 [n° ligne 2]
Dragon 32	LOG (exp)		MID$ (chain, début [longueur])				NEW	NEXT [var ..]		ON var GOSUB n° ligne 1 [n° ligne 2]
Hector (ex Victor)	LOG (exp)	MAX (exp 1, exp 2)	MID$ (chain, début [longueur])	MIN (exp 1, exp 2)			NEW	NEXT [var]	ERROR n° ligne	ON var GOSUB [n° ligne 1 [n° ligne 2 exp 1 [exp 2
Lynx	LN (exp)		MID$ (chain, début [longueur])		VAR = arg 1 MOD arg 2		NEW	NEXT [var]		
MZ 80 A	LN (exp)		MID$ (chain, début [longueur])				NEW	NEXT [var]		ON var GOSUB n° ligne 1 [n° ligne 2]
Oric 1	LN (exp) (LOG = décimal)		MID$ (chain, début [longueur])				NEW	NEXT [var]		ON var GOSUB n° ligne 1 [n° ligne 2]
PET/CBM	LOG (exp)		MID$ (chain, début [longueur])				NEW	NEXT [var]		ON var GOSUB n° ligne 1 [n° ligne 2]
Thomson TO7	LOG (exp)		MID$ (chain, début [longueur])		var = arg 1 MOD arg 2		NEW	NEXT	ON ERROR GOTO n° ligne	ON var GOSUB n° ligne 1 [n° ligne 2]
TI 99	LOG (exp)						NEW	NEXT		ON var GOSUB n° ligne 1 [ligne 2]
TRS 80 modèles 1 et 3	LOG (exp)		MID$ (chain, début, longueur)			NAME ancien nom AS nouveau nom	NEW	NEXT [var]	ON ERROR GOTO n° ligne	ON var GOSUB n° ligne 1 [n° ligne 2]
Vic 20	LOG (exp)		MID$ (chain, début, longueur)				NEW	NEXT [var ..]		ON var GOSUB n° ligne 1 [n° ligne 2]
ZX 81	LN (exp)		chain (début TO fin)				NEW	NEXT var		
ZX Spectrum	LN (exp)		chain (début TO fin)				NEW	NEXT var		

arg. : argument numérique, **exp.** : expression, **chai(n)** : chaîne numérique, **adr.** : adresse, **var** : variable (nom de variable) [ ] : indique une clause optionnelle, { } : indique des clauses équivalentes

Instructions Basic (ON/GOTO à RANDOMIZE)

Ordre Basic (Basicos) / Ordinateur individuel	ON/GOTO (SELON/VATEN) Branchement à une ligne selon la valeur d'une variable	OPEN (OUVRE) Ouverture d'un fichier	OUT (EMET) Envoie la donnée sur le port indiqué	PEEK (IMAGE) Donne le contenu d'une adresse mémoire	PLOT (DESSINE) Allume point des coordonnées indiquées (ou positionne le curseur)	POKE (GARNIS) Range à l'adresse indiquée l'octet spécifié	POS (POS) Positionne le curseur ou donne sa position sur ligne courante	PRINT (ECRIS) Impression des résultats ou écriture dans un fichier (#)	PRINT USING (ECRIS SUIVANT) Formatage	RANDOMIZE (HASARDE) Génère une série de nombres aléatoires
Alice (Tandy MC 10)	ON var GOTO n° ligne 1 [, ligne 2..]			PEEK (adr)	SET (x, y, z)	POKE adr, octet	PRINT @	PRINT exp [PRINT TAB (exp) exp PRINT e exp		
Apple	ON var GOTO n° ligne 1 [, ligne 2..]	OPEN nom de fichier		PEEK (adr)	PLOT x, y	POKE adr, octet	POS (x, y) POS (a) position en ligne courante	PRINT chain var [,]	Usage spécifique pour formatage	
Atari 400	ON var GOTO n° ligne 1 [, n° ligne 2..]	OPEN # n° fiche mode code control, « nom fich »		PEEK (adr)	PLOT x, y	POKE adr, octet	POSITION x, y	PRINT exp PRINT # n° fich enreg [, enreg]		RND (arg)
Atom		var = OPENIN ou (lecture) OPENOUT (écriture) « nom de fichier »		octet = ? adr	MOVE arg 1 arg 2 PLOT	?adr = octet		PRINT [exp]		RND % arg
MC	{ON exp var} GOTO n° ligne 1 [, n° ligne 2]	n° fichier = {OPENIN (lect) / OPENOUT (écrit)}	BPUT # (arg) (BGET operation inverse)	? adr	{MOVE, PLOT 4 x, y (cf manuel PLOT arg)}	? adr, octet	POS position en ligne courante	PRINT exp [exp] PRINT # nom fich, enreg [, enreg]	opérateur de format @	RND (arg)
Commodore 64	ON var GOTO n° ligne 1 [, n° ligne 2..]	OPEN arg {1 cassi, 8 disk / 0 lect, 1 écrit} « nom de fichier »		PEEK (adr)		POKE adr, octet	POS (arg) position en ligne courante	PRINT exp [exp] PRINT # n° fich enreg [, enreg]		RND (TI)
Dai	ON var GOTO n° ligne 1 [, n° ligne 2..]		OUT port, donnée	var = PEEK (adr)	DOT x, y couleur	POKE adr, octet	CURSOR x, y var = CURX position ligne courante	PRINT exp [exp]		RND (arg) N.B. le signe de arg définit la fonct on
Dragon 32	ON var GOTO n° ligne 1 [, n° ligne 2..]	OPEN mode, # n° fich, « nom de fichier »		PEEK (adr)	PSET x, y, couleur (PRESET fonction inverse)	POKE adr, octet	POS position ligne courante	PRINT # arg, exp [, exp] PRINT exp, exp...	Usage spécifique pour formatage	
Hector (ex Victor)	ON var GOTO {n° exp 1 / exp 1} {, exp 2 / exp 2}		OUT port, octet	PEEK (adr)	PLOT x, y [, largeur hauteur] couleur	POKE adr, octet	CURSOR x, y POS (arg) position ligne courante	PRINT arg exp		SEED exp
Lynx			OUT port, donnée	PEEK (adr)	DOT (x, y)	POKE adr, octet	POS position ligne courante	PRINT exp [exp...]		RANDOM
MZ 80 A	ON var GOTO n° ligne 1 [, n° ligne 2..]		OUT port, donnée	PEEK (adr)		POKE adr, octet	MOVE x, y	PRINT T enreg [, enreg] PRINT # arg, var [, var]		RND (-exp)
Oric 1	ON var GOTO n° ligne 1 [, n° ligne 2..]		POKE 300, valeur	PEEK (adr)	CURSET x, y, code N.B. PLOT a une autre signification	POKE adr, octet	CURSOR x, y POS position en ligne courante			RND (arg)
PET/CBM	ON var GOTO n° ligne 1 [, n° ligne 2..]	OPEN arg {1 cass, 8 disk / 0 lect, 1 écrit} « nom de fichier »		PEEK (adr)		POKE adr, octet	POS (arg) position en ligne courante	PRINT exp [, exp] PRINT # n° fich enreg [, enreg]	PRINT USING	RND (TI)
Thomson TO7	ON var GOTO n° ligne 1 [, n° ligne 2..]			PEEK (adr)	PSET	POKE adr, octet	LOCATE x, y	PRINT arg [, exp]	PRINT USING	
TI 99	ON var GOTO n° ligne 1 [, n° ligne 2..]						(autre signification)	PRINT exp [, exp] N.B. DISPLAY équivalent		RANDOMIZE arg
TRS 80 modèles 1 et 3	ON var GOTO n° ligne 1 [, n° ligne 2..]	OPEN « I/O » # n° nom seq. OPEN « R » # n° nom, longueur direct	OUT port, octet	PEEK (adr)	SET (x, y)	POKE adr, octet	POS (e) position en ligne courante	PRINT var PRINT # n° fich enreg [, enreg...]	PRINT USING format var	RANDOM
Vic 20	ON var GOTO n° ligne 1 [, n° ligne 2..]	OPEN exp n° fich, mode « nom fich »		PEEK (adr)		POKE adr, octet	POS (a) position en ligne courante	PRINT [« mess »] [var] PRINT # n° fich enreg [, enreg...]		RND (TI)
ZX 81				PEEK (adr)	PLOT x, y	POKE adr, octet	PRINT AT x, y	PRINT [« message »] [var]		RAND (arg)
ZX Spectrum		OPEN # n° fichier	OUT port, octet	PEEK (adr)	PLOT x, y	POKE adr, octet	PRINT AT x, y	PRINT [« message »] [var...]		RAND (arg)

arg. : argument numérique **exp.** : expression **chai(n)** : chaîne numérique **,adr** : adresse ; **var** = variable (nom de variable) **[,]** : indique une clause optionnelle **{ }** : indique des clauses équivalentes

Instructions Basic (READ à SAVE)

Ordre Basic (Basicos) / Ordinateur individuel	READ (EMPLIS) Lecture d'une liste DATA	REM (REM) Insertion de lignes commentaires	RENUM (RENUM) Renumérotation des lignes de programme	RESTORE (RESTAURE) Replacement en début de liste DATA	RESUME (REPRENDS) Retour juste avant le débranchement d'erreur	RETURN (REVIENS) Fin de sous programme retour dernière GOSUB	RIGHT$ (DROITE$) Donne les caractères de droite d'une chaîne	RND (HSD) Tirage d'un nombre aléatoire	RUN (EXE) Exécution d'un programme	SAVE (COPIE) Sauvegarde d'un programme
Alice (Tandy MC 10)	READ var [, var]	REM commentaire		RESTORE		RETURN	RIGHT$ (char, longueur)	RND (arg)	RUN	
Apple	READ var [, var]	REM commentaire		RESTORE	RESUME	RETURN	RIGHT$ (chain, longueur)	RND (arg)	RUN [n° ligne]	SAVE [« nom fich »]
Atari 400	READ var [, var]	REM commentaire		RESTORE n° ligne		RETURN	A$ (1 10)	RNO (arg)	RUN	SAVE « nom fichier » (cassette cf CSAVE)
Atom	READ	REM commentaire				RETURN	chain + longueur	RND arg	RUN	SAVE nom fichier
BBC	READ var [, var]	REM commentaire	RENUMBER [début] [, pas]	RESTORE (exp)	ON ERROR OF	RETURN	RIGHT$ (chain, longueur)	RND (arg)	RUN	SAVE {[« nom fiche »] / [chain]}
Commodore 64	READ var [, var]	REM commentaire		RESTORE		RETURN	RIGHT$ (chain, longueur)	RND (arg)	RUN [n° ligne]	SAVE « nom fichier » # (cassette, cf CSAVE)
Dai	READ var [, var]	REM commentaire		RESTORE		RETURN	RIGHT$ (chain, longueur)	RND (arg)	RUN	SAVE {[chain] / [« nom fich »]}
Dragon 32	READ var [, var]	REM commentaire	RENUM [n° ligne début pas]	RESTORE		RETURN	RIGHT$ (chain, longueur)	RND (arg)	RUN [n° ligne]	
Hector (ex Victor)	READ var [, var]	REM commentaire	RENUM [n° ligne] [, pas]	RESTORE exp [n° ligne]		RETURN	RIGHT$ (chain, longueur)	RND (min, max)	RUN [n° ligne]	SAVE [« nom fich »]
Lynx	READ var [, var]	REM commentaire	RENUM [n° ligne début, pas]	RESTORE		RETURN	RIGHT$ (chain, longueur)	RND ou RAND (arg)	RUN n° ligne	SAVE « nom fichier » [« ou la n° ligne]
MZ 80 A	READ var [, var]	REM commentaire		RESTORE (arg)	RESUME {NEXT / n° ligne}	RETURN	RIGHT$ (chain, longueur)	RND (arg)	RUN	SAVE [FD n° drive] « nom fichier »
Oric 1	READ var [, var]	REM commentaire		RESTORE		RETURN	RIGHT$ (chain, longueur)	RND (arg)	RUN [n° ligne]	
PET/CBM	READ var [, var]	REM commentaire	RENU [ligne 1 ligne 2, pas]	RESTORE		RETURN	RIGHT$ (chain, longueur)	RND (arg)	RUN [n° ligne]	SAVE « nom fichier » # (cassette cf CSAVE)
Thomson TO7	READ var [, var]	REM commentaire		RESTORE		RETURN	RIGHT$ (chain, longueur)	RND	RUN [n° ligne]	SAVE « nom fich »
TI 99	READ var [, var]	REM commentaire	RES [n° ligne début pas]	RESTORE		RETURN		RND	RUN	SAVE « nom fichier »
TRS 80 modèles 1 et 3	READ var [, var]	REM commentaire	RENUM n° ligne début, pas	RESTORE	RESUME [n° ligne]	RETURN	RIGHT$ (chain, longueur)	RND (arg)	RUN [n° ligne]	SAVE « nom fich. » (cassette, cf CSAVE)
Vic 20	READ var [, var]	REM commentaire		RESTORE		RETURN	RIGHT$ (chain, longueur)	RND (arg)	RUN (N° ligne)	SAVE « nom fich. » # (cassette cf CSAVE)
ZX 81		REM commentaire				RETURN	chain (début TO)	RND	RUN {[n° ligne / var / exp]}	
ZX Spectrum	READ var [, var...]	REM commentaire		RESTORE {[n° ligne] / [exp]}		RETURN	chain (début TO)	RND	RUN {[n° ligne / var / exp]}	

arg. : argument numérique, **exp.** : expression, **chai(n)** : chaîne alphanumérique **adr** : adresse **var** : variable (nom de variable) **[,]** : indique une clause optionnelle **{ }** : indique des clauses équivalentes

Instructions Basic (SCRN à SWAP)

Ordre Basic (Basicois) / Ordinateur individuel	SCRN (ECRAN) indique la couleur du point de coordonnées spécifiées	SEG$ (EXTR$) Extrait une chaîne d'un plus grande	SGN (SGN) Signe	SIN (SIN) Sinus	SPC (ESP) Génère une chaîne d'espaces	SQR (RAC) Racine carrée	STOP (ARRET) Arrêt d'exécution	STRING$ (CHAINE$) Génère une chaîne	STR$ (TXT$) Change une expression numérique en une chaîne	SWAP (ECHANGE) Echange le contenu de deux variables
Alice (Tandy MC 10)	POINT (x, y)		SGN (exp)	SIN (exp)		SQR (exp)	STOP		STR$ (arg)	
Apple	SCRN (x y)		SGN (exp)	SIN (exp)	SPC (arg) (utilise avec PRINT)	SQR (exp)	STOP		STR$ (arg)	
Atari 400			SGN (exp)	SIN (exp)		SQR (exp)	STOP		STR$ (arg)	
Atom			SGN (exp)	SIN (exp)		SQR (exp)			STR$ (arg)	
BBC	POINT (x y)		SGN (exp)	SIN (exp)	SPC (arg) (utilise avec PRINT)	SQR (exp)	STOP	STRING$ (longueur, chain)	STR$ (arg)	
Commodore 64			SGN (exp)	SIN (exp)	SPC (arg) (utilise avec PRINT)	SQR (exp)	STOP		STR$ (arg)	
Dai	SCRN x y		SGN (exp)	SIN (exp)	SPC (arg)	SQR (exp)	STOP		STR$ (arg)	
Dragon 32			SGN (exp)	SIN (exp)		SQR (exp)	STOP	STRING$ (longueur, chain)	STR$ (arg)	
Hector (ex Victor)	POINT (x y)		SGN (exp)	SIN (exp)	SPC (arg)	SQR (exp)	STOP		STR$ (arg)	SWAP var 1 var 2
Lynx			SGN (exp)	SIN (exp)		SQR (exp)	STOP		STR$ (arg)	SWAP var 1 var 2
MZ 80 A			SGN (exp)	SIN (exp)		SQR (exp)	STOP		STR$ (arg)	
Oric 1	SCRN (x, y)		SGN (exp)	SIN (exp)	SPC (arg)	SQR (exp)	STOP		STR$ (arg)	
PET/CBM			SGN (exp)	SIN (exp)	SPC (arg) (utilise avec PRINT)	SQR (exp)	STOP		STR$ (arg)	autre signification
Thomson TO7	POINT (x y)		SGN (exp)	SIN (exp)	SPC (arg)	SQR (exp)	STOP		STR$ (arg)	
TI 99		SEG$ (chain, arg 1 arg 2)	SGN (exp)	SIN (exp)		SQR (exp)	STOP		STR$ (arg)	
TRS 80 modèles 1 et 3			SGN (exp)	SIN (exp)	SPACES (arg)	SQR (exp)	STOP	STRING$ (longueur, chain)	STR$ (arg)	SWAP var 1 var 2
Vic 20			SGN (exp)	SIN (exp)	SPC (arg) (utilise avec PRINT)	SQR (exp)	STOP		STR$ (arg)	
ZX 81			SGN (exp)	SIN (exp)		SQR (exp)	STOP		STR$ (arg)	
ZX Spectrum	ATTR (x, y)		SGN (exp)	SIN (exp)		SQR (exp)	STOP		STR$ (arg)	

arg. : argument numérique **exp.** : expression **chai(n)** : chaîne numérique **adr.** : adresse **var** : variable (nom de variable) , [,] : indique une clause optionnelle ; { } indique des clauses équivalentes

Instructions Basic (SYSTEM à WIDTH)

Ordre Basic (Basicois) / Ordinateur individuel	SYSTEM (SYS) Quitte le Basic pour revenir dans le système d'exploitation	TAN (TAN) Tangente	TEXT (TEXTE) Passage en mode TEXTE	TSOFF (TROFF) Annule TRON	TSON (TRON) Visualise, en exécution, le n° de lignes exécutées	USR (USR) Transmet une valeur à un programme en langage machine	VAL (NBRE) Donne la valeur numérique d'une chaîne	WAIT (ATTENDS) Suspend un programme pour un temps donné	WHILE/WEND (TANT QUE/FAIRE) Tant que faire	WIDTH (LONGUEUR) Définit la longueur max. des lignes imprimées
Alice (Tandy MC 10)		TAN (exp)					VAL (chain)			
Apple		TAN (exp)	TEXT	NOTRACE	TRACE	USR (param.)	VAL (chain)	WAIT adr. arg 1 [arg 2]		POKE 32 val POKE 33 val
Atari 400	BYE		GRAPHICS arg			USR (param.)	VAL (chain)			POKE 82 val POKE 83 val
Atom		TAN exp				LINK adr	VAL (chain)	WAIT 1/50e	DO ordre UNTIL condition (log que inverse)	WIDTH val
BBC	*DISK	TAN (exp)	MODE arg	TRACE OFF	TRACE ON	USR (param.)	VAL (chain)		REPEAT ordre UNTIL condition	WIDTH val
Commodore 64		TAN (exp)				USR (param.) (cf. manuel SYS)	VAL (chain)	WAIT adr arg 1 arg 2	WHILE condit ons ordres WEND	
Dai		TAN (exp)	MODE (n)	TROFF	TRON		VAL (chain)	WAIT {MEM} adr arg 1 arg 2 WAIT TIME arg		
Dragon 32		TAN (exp)		TROFF	TRON	USR (param.)	VAL (chain)			
Hector (ex Victor)		TAN (exp)				USR adr [registre] [param.]	VAL (chain)			
Lynx		TAN (exp)	Autre utilisation	TRACE OFF	TRACE ON		VAL (chain)	PAUSE exp (secondes)	WHILE condition WEND ordre	
MZ 80 A	BYE	TAN (exp)				USR (param.)	VAL (chain)			
Oric 1		TAN (exp)	TEXT	TROFF	TRON	USR (param.) & (param.)	VAL (chain)	WAIT arg NB arg en 100e de seconde	REPEAT ordre UNTIL exp	POKE 46 val
PET/CBM		TAN (exp)		OFF	TRACE	USR (param.) (cf manuel SYS)	VAL (chain)	WAIT adr arg 1, arg 2	WHILE condition ordres WEND	
Thomson TO7		TAN (exp)		TROFF	TRON	USR (param.)	VAL (chain)			
TI 99		TAN (exp)		UNTRACE	TRACE		VAL (chain)			
TRS 80 modèles 1 et 3	SYSTEM	TAN (exp)		TROFF	TRON	USR (param.)	VAL (chain)	WAIT n° port masque sélection		
Vic 20		TAN (exp)				USR (param.)	VAL (chain)	WAIT adr. arg 1 arg 2		WIDTH val
ZX 81		TAN (exp)				USR (adr) (cf CALL)	VAL (chain)	PAUSE arg		
ZX Spectrum		TAN (exp)				USR (adr) (cf CALL)	VAL (chain)	PAUSE arg		

arg. : argument numérique **exp.** : expression **chai(n)** : chaîne numérique **adr.** : adresse , **var** : variable (nom de variable) [,] : indique une clause optionnelle , { } indique des clauses équivalentes

libérez votre vocabulaire grâce à une nouvelle science : la logotronique

et un programme « académique » pour l'étudier

La logotronique s'est imposée tout récemment comme une des branches montantes des sciences du langage. Nous sommes heureux d'ouvrir aujourd'hui nos colonnes aux deux « pères » du Logotron, dont nous révélons, en exclusivité pour nos lecteurs, le fonctionnement.

La logotronique est l'étude expérimentale de la stabilité des formes linguistiques générées par ordinateur. C'est un sous-ensemble de la littératronique, dont l'inventeur est le professeur Escarpit, que nos lecteurs connaissent certainement.

Le thème général de la littératronique est la synthèse du langage : poèmes, romans, discours techniques, philosophiques, ou même politiques (voir l'excellent ouvrage du professeur Escarpit, paru il y a quelques années : *Le Littératron*.

Première parution : L'OI n° 18

Plus modestement, la logotronique se propose de créer des mots nouveaux. Le monde actuel est en pleine mutation, et la demande en mots nouveaux se fait chaque jour plus vive.

Les partisans de cette discipline de pointe de la linguistique pensent qu'il est grand temps de faire éclater le « carcan du dictionnaire ». En effet, ces mots du dictionnaire, il a bien fallu, au fil des siècles, qu'on les crée. Et on ne voit pas pourquoi nous accepterions un figement brutal de la langue, bien modestement enrichie par les sporadiques contributions de l'Académie.

Le logotron est donc un vaste programme de synthèse lexicologique. Le programme donné en annexe de cet article *(page 29)* peut générer 68 000 mots. Et ce n'est qu'un début. Il est apparu, de fait, que nous vivions à la surface d'un véritable gisement sémantique à peu près inexploité, et que nos travaux ont mis à jour tout récemment.

Il reste, bien entendu, à trouver un sens à ces mots produits par le logotron. L'équipe de l'Institut de Logotronique de la Faculté des Lettres d'Aix-en-Provence s'y emploie, et on trouvera ci-après un échantillon de mots, pourrait-on dire « prêts à l'emploi ».

La logotronique s'adresse à tous les secteurs du langage, mais, d'ores et déjà, un fonctionnement thématique est envisageable.

Les débouchés sont considérables et, à une époque où la France équilibre à grand peine sa balance des paiements, nous pouvons dire qu'étant donné l'avance prise, nous sommes très bien placés sur le marché de l'exportation.

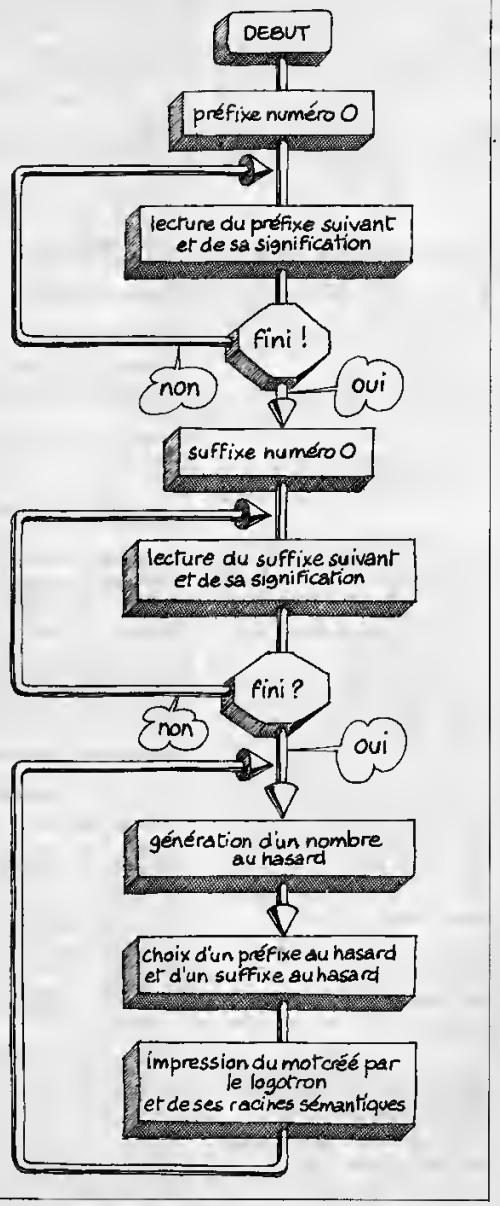
La logotronique prendra-t-elle une place dans notre vie de chaque jour ? Grâce à l'avènement des ordinateurs individuels, nous pouvons répondre : OUI. Nous avions les calculateurs de poches, les ordinateurs programmables de poche, les machines à traduire de poche. Dans un avenir que nous espérons proche, nous aurons dans la main des UL (unités de lexicosynthèse) qui, sous une simple pression de touche, fourniront toute une gamme de mots propres à enrichir notre discours, notre conversation.

Résultat des premières recherches

Voici un bref échantillon des mots qui portent maintenant le label de l'Institut de Logotronique.

Ambulochrone : qui a la faculté de pouvoir se déplacer dans le temps.
Bibliostat : presse-livre.
Bactériomancie : prévision de l'avenir en regardant dans un microscope.
Cryotropisme : afflux saisonnier dans les stations de sports d'hiver.
Chronothérapeuthe : médecin qui soigne en laissant faire le temps.
Cosmophobe : se dit de quelqu'un qui ne peut pas supporter l'Univers.
Cératocéphale : cocu.
Chronophage : personne qui vous prend votre temps (mangeuse de temps).
Cosmogène : se dit d'une situation où l'Univers ne demande qu'à apparaître.
Cryptodyne : véhicule dont le mode de propulsion est caché (soucoupe volante).
Cosmorrhée : création divine.
Elastolithe : variété de pierre élastique.
Elastosaure : ancêtre du ver de terre.
Episcodrome : terrain d'atterrissage pour évêques.
Hypolithe : sous-pierre.
Hémiscope : lunettes dont un verre est opaque, pour ne regarder qu'à moitié.
Hémigame : à moitié marié.
Héliorragie : éruption solaire.
Logomètre : pour discours mesurés.
Logothérapie : psychanalyse.
Logostase : quand on n'a plus rien à dire.
Latérograde : personne marchant à côté de ses chaussures.
Logotome : pour faire des phrases hachées.
Micoclaste : casseur de champignons.
Monoscope : petit cylindre en matériau quelconque, pour ne regarder qu'une seule chose à la fois.
Mésographe : pour écrire entre les lignes.
Nécrogramme : faire-part de décès.
Pornotropisme : goût pour la littérature porno.
Pseudothérapeuthe : psychanalyste.
Panmanie : manie de tout.
Panscaphe : véhicule pour aller n'importe où.
Pseudopathe : personne qui n'est pas vraiment malade.
Phallostat : slip.
Paléogame : qui se reproduit de façon désuète.
Podoclaste : casse-pieds.
Pangamie : tendance à épouser tout le monde.
Parallélioscaphe : véhicule pour univers parallèles.
Panphobie : horreur de tout.
Pyrotope : cheminée.
Spéléosophie : sagesse des profondeurs.
Statodynamique : étude de l'évolution des états stationnaires.
Théodyne : machine marchant par l'opération du Saint-Esprit.
Théotope : paradis.

Organigramme



Dans le but de ne pas freiner les voies sur lesquelles progresse la recherche l'organigramme présenté ci dessus ne comporte effectivement pas de fin

Mettez-vous à la logotronique

Le programme logotron est tres simple Il est constitue de deux tables de DATA La première table contient des débuts de mots, qui doivent se terminer par une voyelle Exemple aero, latero, hydro, etc La seconde contient les fins de mots On les combine aleatoirement Les donnees sont entrées par couples car le programme fournira en même temps la signification etymologique des mots creés Exemple clepto, dérober...

Et, dans la seconde table gène, formation

Les DATA sont donc lues par couples 90 READ X$, Y$

On constitue quatre tables P$ (I) sont les debuts de mots S$ (J) sont les fins de mots PO$ (I) et SO$ (J) sont les significations de ces deux racines empruntees au grec ou au latin Le programme a été ecrit pour un Apple II L instruction 160 donne la racine du générateur de nombres aleatoires

Si on dispose d'une imprimante, on peut sortir immediatement une suite de mots (logorrhee) Le reste est litterature

On pourrait fabriquer un logotron avec simplement deux boîtes contenant des cartons, et en piquant dans ces boîtes au hasard Mais c'est un peu lent Seul l ordinateur est a même de produire un tel déferlement de mots inconnus

Une science déjà officialisée par la télé

Le premier mars 1980, nous avons presenté le logotron, ne lors d un apres-midi pluvieux, à l'emission Temps X, sur TF1 Pour l occasion, nous avions mis les mots du paragraphe 2 et leur signification dans un autre secteur de la memoire Le programme incluait aleatoirement ces mots dans sa production, et un FLASH les signalait au telespectateur Ceci etait alors interprété comme une entrée dans le « champ sémantique » de l appareil qui, apres « activation de son module sémantique », « calculait » la signification Le tout agrémente de clicks simulant une imprimante inexistante, et de gargouillis musicaux du meilleur effet

Nous fûmes assez effrayés par le courrier reçu suite à cette emission Si l on excepte une lettre d Escarpit, ravi de trouver là l'illustration de son roman picard, *Le Littératron,*

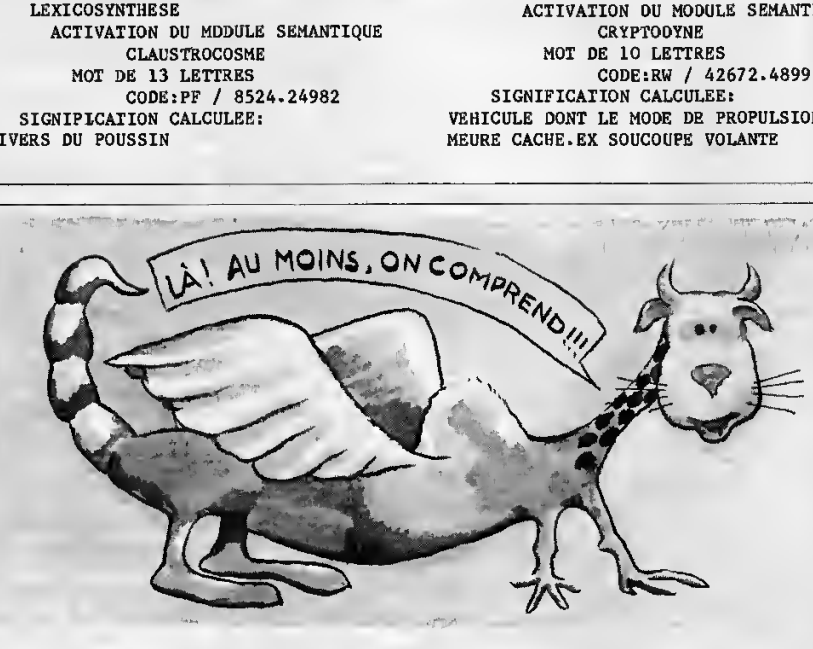
La liste du programme

```
100  REM  PROGRAMME LOGOTRON
110  REM  AUTEURS : JEAN-PIERRE PETIT ET MARC ARONDEL
120  REM  COPYRIGHT L'ORDINATEUR INDIVIDUEL ET LES AUTEURS
130  TEXT
140  SPEED= 50
150  DIM P$(400), DIM PO$(400)  DIM S$(400)  DIM SO$(400)
200  REM  -----
210 M = 0  REM  INDICE PREFIXE
220  READ X$,Y$
230  IF X$ = "FIN" THEN 310
240 M = M + 1
250 P$(M) = X$ PO$(M) = Y$
260  GOTO 220
300  REM  -----
310 N = 0  REM  N INDICE SUFFIXES
320  READ X$,Y$
330  IF X$ = "FIN" THEN 410
340 N = N + 1
350 S$(N) = X$ SO$(N) = Y$
360  GOTO 320
400  REM  -----
410  HOME  HTAB 15: VTAB 15: PRINT "***LOGOTRON***"  PRINT  PRINT
420  HTAB 8: PRINT "PAR J.-P. PETIT ET MARC ARONDEL"  PRINT  PRINT
430  PRINT  PRINT  HTAB 9: PRINT "LE NOMBRE DE MOTS EST ";M * N
440  PRINT . HTAB 9: PRINT "FACULTE DES LETTRES D'AIX
450  PRINT  HTAB 13: PRINT "EN PROVENCE 13100"
460  PRINT
500  REM  -------- BOUCLE D'IMPRESSION ------
510 RD = ( - 256 *  PEEK (79) -  PEEK (78))
520 I =  RND (1) * M + 1 J =  RND (2) * N + 1
530  PRINT P$(I) + S$(J)  PRINT
540  HTAB 5
550  PRINT SO$(J);"-";PO$(I)
560  PRINT  GOTO 510
1000  REM  -----
1010  REM  ******* LISTE DES PREFIXES *****
1020  DATA    ACRO,EXTREMITE,ADENO,GLANDE,AERO,AIR,ADNO,CHAMP,ALGO,DOU
      LEUR,ALLO,AUTRE,ANDRO,HOMME,ARTHRO,NOMBRE,ARRHRO,HALE
1030  DATA ARSENO,VIRIL,ANTE,AVANT,ANTI,CONTRE,ANTHO,FLEUR,ARTERIO,AR
      TERE,ARTHRO,ARTICULATION
1040  DATA ASTRO,ETOILE,ASTRO,ASTRE,ASTHENO,FATIGUE,ANTHROPO,HOMME,A
      PO,A FAMILE DE,APHRO,MOUSSE,ARCHEO,ANCIEN,ARGO,BRILLANT
1050  DATA  BACTERIO,BACTERIE,BALANO,GLANDE,BARO,PESANTEUR,BARO,PRESS
      ION,BARY,LOURD,BATHY,PRESSION,BIBLIO,LIVRE,BIDULO,MACHIN
1060  DATA    BIO,VIE,BLENO,MUCUS,BLASTO,GERME,BOLO,JETER,BRACHY,COUR
      T,BRADY,LENT,BRONCHO,BRONCHE,BRONTO,TONNERRE,BUTYRO,BEURRE
1070  DATA  CACO,MAUVAIS,CARDIO,COEUR,CARYO,NOIX,CATA,SUS,CENTRO,CENT
      RE,CERCO,QUEUE,CINEMATO,MOUVEMENT
1080  DATA CLEITO,DEBORER,CLIMATO,REGION,CLINO,INCLINE,CENO,VIDE,CEPH
      ALO,TETE,CERATO,CORNE,CHEIRO,MAIN,CHOLI,PINCE,CHRIO,CRINIERE
1090  DATA CHIMIO,CHIMIE,CHLORO,JAUNE,CHRYSO,OR,CHROMO,COULEUR,CHRONO
      ,TEMPS,CO,AVEC,COELO,CREUX,COPRO,EXCREMENT,CONTRA,CONTRE
1100  DATA  COSMO,COSMIQUE,CRANIO,CRANE,CRISTALLO,VERRE,CRYO,FROID,CR
      YPTO,CACHE,CYANO,BLEU,CYCLO,CERCLE,CYNO,CHIEN,CYPHO,CONVEXE,CYTO
      ,CELLULE
1110  DATA    DACTYLO,DOIGT,DECI,DIXIEME,DERMATO,PEAU,DI,DOUBLE,DIA,A
      TRAVERS,DIPLO,DOUBLE,DINO,TERRIBLE,DECA,DIX,DEMO,PEUPLE,DEMONO,
      DIABLE,DENDRO,ARBRE
1120  DATA  DEXTRO,DEXTRE,DEUTERO,SECOND,DENTRO,DROIT,DODECA,DOUZE,DOXO,
      OPINION,DRAMA,THEATRE,DROMO,COURSE,DYNAMO,ENERGIE,DYS,DIFFICULTE
      ,ECTO,A L'EXTERIEUR,ELASTO,ELASTIQUE
1130  DATA    ELECTRO,ELECTRICITE,EMBRYO,EMBRYON,ENCEPHALO,CERVEAU,END
      O,A L'INTERIEUR,ENTERO,ENTRAILLES,ENTOMO,INSECTE,EO,AURORE,EPI,S
      UR,EPISTEMO,SCIENCE
1140  DATA    ERGO,TRAVAIL,ESTHETICO,ESTHETIQUE,EROTICO,EROTISME,ETHYMO
      ,VRAI,EXO,AU DEHORS,EXTRA,AU DEHORS,FANTASMO,ILLUSION,FIBRO,FIBR
      E,GALACTO,LAIT,GAMO,MARIAGE,GASTRO,VENTRE
1150  DATA  GEO,TERRE,GERONTO,VIEILLARD,GENO,RACE,GLOSSO,LANGUE,GLYCO
      ,SUCRE,GNOSEO,CONNAISSANCE,GONIO,ANGLE,GONO,SEMENCE,GRAMMO,MESSA
      GE,GRAPHO,ECRITURE,GRAVITO,GRAVITE,GYMNO,NU
1160  DATA    GYNO,FEMME,GYRO,CERCLE,HALO,SEL,HECTO,CENT,HELIO,SOLEIL,H
      EMATO,SANG,HEMI,A MOITIE,HEPATO,FOIE,HEPTA,SEPT,HETERO,AUTRE,HEX
      A,SIX
1170  DATA  HIERO,SACRE,HIPPO,CHEVAL,HISTO,TISSU,HODO,CHEMIN,HOLO,ENT
      IER,HOMEO,SEMBLABLE,HOMO,SEMBLABLE,HYDRO,EAU,HYGRO,HUMIDITE,HYLO
      ,MATIERE
1180  DATA  HYMENO,MEMBRANE,HYPER,AU DELA DE,HYPNO,TISSU,HYPO,INSUFFI
      SANCE,HYSTERO,UTERUS,ICHTIO,POISSON,ICONO,IMAGE,IDEO,IDEE,IDIO,P
      ROPRE
1190  DATA  INFLATO,GONFLER,INFRA,EN DESSOUS,ISO,EGAL,INTRA,EN DEDANS
      ,JUXTA,A COTE,KILO,MILLE,LATERO,COTE,LARYNGO,GORGE,LEPTO,MINCE,L
      EVO,GAUCHE
1200  DATA  LITO,GRAISSE,LITHO,PIERRE,LOGO,DISCOURS,LOXO,OBLIQUE,MACR
      O,GRAND,MAGNETO,MAGNETISME,MASTO,MAMELLE,MEGE,GRAND,MEGALO,GRAND
      ,MELO,CHANT,MERO,PARTIE,MESO,MILIEU
1210  DATA  METRO,RELEVE DANS LES AIRS,META,CHANGEMENT,METRO,MESURE,MI
      CRO,PETIT,MISO,HATE,MNEMO,MEMOIRE,MORPHO,FORME,MYCO,CHAMPIGNON,M
      YELO,MOELLE,MYO,MUSCLE
1220  DATA  MYTHO,MYTHE,NECRO,MORT,NEO,NOUVEAU,NEURO,NERF,NEMATO,FIL,
      NEVRO,NERF,NOO,ESPRIT,NOMO,LOI,NOSO,MALADIE,NYCTO,NUIT,NUCLEO,NO
      YAU,OCTO,HUIT,ODO,ROUTE,OLEO,HUILE
1230  DATA  OLIGO,PEU NOMBREUX,OMNI,TOUT,ONYO,L'ETRE,OO,OEUF,OPNTALMO
      ,OEIL,ORCHIDO,COUILLE,ORGANO,ORGANE,ORNITHO,OISEAU,ORO,MONTAGNE,
      ORTHO,DROIT,OSCILLO,OSCILLATION,OSMO,PULSION
1240  DATA  OTO,OREILLE,OVO,OEUF,PACHY,EPAIS,PALEO,ANCIEN,PAN,TOUT,PA
      NTO,TOUT,PARA,VOISIN DE,PATHO,PATHE,PARALLELO,PARALLELE,PATA,P
      ATA,PATHO,SOUFFRANCE,PEDO,PIED,PEDO,SOL,PENTA,CINQ
1250  DATA  PERI,AUTOUR DE,PETRO,PIERRE,PHAGO,MANGER,PHALLO,ZIZI,PHAR
      MACO,VISIBLE,PHARMACO,MEDICAMENT,PHARINGO,GOSIER,PHREN,APPARAITRE
      ,PHILO,AIMER,PHOBO,PEUR DE,PHONO,SON
1260  DATA  PHOTO,LUMIERE,PHRENO,ESPRIT,PHYLO,RACE,PHYSIO,NATURE,PHYT
      O,PLANTE,PINACO,TABLEAU,PITHECO,SINGE,PLASMO,FACONNER,PLASTO,MOD
      ELAGE,PLECTO,SOUDER,PLEO,PLUS
1270  DATA  PLEISTO,BEAUCOUP,PLESSIO,PROCHE,PLEURO,A COTE,PLOUTO,RICHE
      ,PLURI,PLUSIEURS,PNEUMO,POUMON,PODO,PIED,POLY,NOMBREUX,POEMO,COC
      HON,POST,APRES,PRIMO,PREMIER,PRO,DEVANT,PROTO,PREMIER,PSEUDO,FAU
      X
1280  DATA  PSYCHO,AME,PTERO,AILE,PYO,PUS,PYRO,FEU,QUASI,PRESQUE,RADI
      O,RAYON,RETRO,EN ARRIERE,RHEO,COULER,RHINO,NEZ,RHIZO,RACINE,RHOM
      BO,LOSANGE,SCATO,EXCREMENT,SCLERO,DUR,SELENO,LUNE,SEMIO,SENS,SEM
      A,SIGNE,SEMI,A DEMI
1290  DATA  SEXO,SEXE,SCHIZO,FENDRE,SIDERO,FER,SIMILI,SEMBLABLE,SOLEN
      O,TUYAU,SOMATO,CORPS,SONO,SON,SPELEO,CAVERNE,SPHENO,COIN,SPHERO,
      SPHERE,SPLENO,RATE,SPONDYLO,VERTEBRE,STAPHILO,GRAIN DE RAISON
1300  DATA  STERCO,TOIT,STENO,STROIT,STERRO,SOLIDE,STERNO,POITRINE,STO
      MATO,BOUCHE,STROBO,TOURBILLON,STYLO,COLONNE,SUE,SOUS,SULFARO,SOU
      FRE,SYL,AVEC,SYM,ENSEMBLE,SYN,AVEC
1310  DATA  TACHY,VITESSE,TACHYTO,VITESSE,TAUTO,LE MEME,TAXO,ORDRE,TE
      CHNO,ART,TELEO,LOIN,TERATO,MONSTRE,TETRA,QUATRE,THALASSO,MER,THE
      O,DIEU,THERMO,CHALEUR,TOMO,COUPER,TONI,TON,TOPO,LIEU,TRIBO,FROTT
      ER,TROPO,TENDANCE VERS,TYPO,CARACTERE
1320  DATA  ULTRA,AU DELA,URANO,CIEL,URO,URINE,VICE-,A LA PLACE DE,VI
      DEO,VOIE,VISIO,VISION,XENO,ETRANGER,XYLO,BOIS,ZOO,ANIMAL,FIN,
1330  REM  -------- FIN DES PREFIXES --------
1340  REM  ****** LISTE DES SUFFIXES *******
1350  DATA    ACRE,EXTREMITE,ACOGLE,QUI CONDUIT,ACOGUE,QUI CONDUIT,AL
      GIE,DOULEUR
1360  DATA    ARQUE,COMMANDER,ANTHE,FLEUR,ASTERE,ETOILE,ASTRE,ETOILE,
      ASTHENIE,FATIGUE,ANTHROPE,HOMME,ARCHIE,COMMANDER,ARGE,BRILLANT
1370  DATA    BACTERIE,BACTERIE,BARE,PRESSION,BATHE,PROFOND,BIE,VIE,BL
      ASTE,GERME,BOLE,JETER,CARDE,COEUR,CARPE,PAUME
1380  DATA  CENTRIQUE,CENTRE,CERE,CORNE,CERQUE,QUEUE,CINESE,MOUVEMENT
      ,CINETIQUE,MOUVEMENT,CLINE,PENTE,CNLE,TUMEUR,CENE,VIDE,CEPHALE,T
      ETE
1390  DATA    CHIMIE,CHIMIE,CHLYSE,OR,CHROME,COULEUR,CHRONE,TEMPS,CNTO
      NE,TERRE,CLASTE,BRISEE,COQUE,GRAINE,COSME,COSMOS,COSMIQUE,COSMOS

1400  DATA  CRATE,FORCE,CULTEUR,CULTURE,CYCLE,CERCLE,CYTE,CELLULE,DRC
      TYLE,DOIGT,DERME,PEAU,DIDACTE,INSTRUIRE,DESE,REUNIR
1410  DATA  DORE,OPINION,DRAME,THEATRE,DROME,COURIR,DROMIE,COURSE,DON
      TE,DENT,DUC,CONDUIRE,DYNE,ENERGIE
1420  DATA  ECTOMIE,ABLATION,EDRE,BASE,ELASTIQUE,ELASTIQUE,ELECTRIQUE
      ,ELECTRICITE,EMIE,EMBRYON,EMIE,SANG
1430  DATA  FANTASME,ILLUSION,FUGE,FUIE,FIBRE,FIBRE,GAME,MARIAGE,GAST
      RE,VENTRE,GASTRE,VENTRE,GEE,TERRE,GENESE,FORMATION,GENE,FORMATI
      ON
1440  DATA    GLOTTE,LANGUE,GLYPHE,GRAVURE,GNATHE,MACHOIRE,GNOSE,CONNA
      ISSANCE,GNOSE,CONNAISSANCE,GONE,ANGLE,GRADE,MARCHER,GRAMME,MESSA
      GE,GRAPHE,ECRIRE
1450  DATA  GRAVITATIONNEL,GRAVITE,GYNE,FEMME,GYRE,CERCLE,HELIE,SOLE
      IL,HEPATIQUE,FOIE,HYDRE,EAU,IATRE,MEDECIN,IDE,FORME
1460  DATA  INFLATION,GONFLEMENT,INFORMATIQUE,INFORMATIQUE,LATRIE,COT
      E,LATRE,ADORATEUR,LITRE,PIERRE,LOGIE,DISCOURS SUR,LOGUE,SPECIALI
      STE DE,LYSE,COUPURE DE,MACHIE,COMBAT,MAGNETIQUE,MAGNETIQUE,MANE,
      FOU DE,MANCIE,PREDICTION DE L'AVENIR PAR
1470  DATA    MANIE,FOLIE DE,MATHEMATIQUE,MATHEMATIQUE,MATIQUE,NOUVEAU
      NE,MATE,QUI SE MEUT,MEGALIE,GRAND,MERE,PARTIE,METRE,MESURE,MNESI
      E,MEMOIRE,MORPHE,FORME
1480  DATA  MORPHISME,FORME,MYCTE,CHAMPIGNON,MYCOSE,CHAMPIGNON,MYTHE,
      MYTHE,MAUTE,QUI NAVIGUE,NESIE,ILE,NEVROSE,NERF,NOME,LOI,NOMIE,LO
      I,NYME,NOM,NUCLEIQUE,NOYAU
1490  DATA  PATHE,SOUFRANCE,PODE,PIED,PETE,QUI VA VERS,PHAGE,MANGEUR
      DE,PHANIE,APPARITION,PHASE,APPARITION D'UNE ETOILE,PHILE,QUI AIM
      E,PHORE,QUI DETRUIT,PHONE,SON,PHONE,QUI PORTE
1500  DATA  PHOTE,LUMIERE,PHRENE,ESPRIT,PHYLAXIE,GARDER,PHYLE,RAMEAU,
      PHYLLE,FEUILLE,PHYSE,GLANDE,PHYSIQUE,PHYSIQUE,PHYTE,PLANTE,PITHE
      QUE,SINGE,PLASME,FACONNER
1510  DATA    PLASTE,MODELAGE,PLASTIE,MODELAGE,PLASTIQUE,MODELAGE,PNEM
      ,RESPIRATION,POIETIQUE,ACTION DE FABRIQUER,POIE,FIRO,POLE,VILLE,
      POLITIQUE,POLITIQUE
1520  DATA  PROPULSION,PROPULSION,PSYCHOLOGIE,PSYCHOLOGIE,PTERE,AILE,
      RADIAL,RAYON,REACTION,REACTION,RHEE,ECOULEMENT,SAURE,LEZARD,SCE
      IPTEUR,QUI ECRIT
1530  DATA  SEXUEL,SEXUEL,SCAPHE,BARQUE,SCOPE,REGARDER,SOCIOLOGIE,SOC
      IOLOGIE,SOLAIRE,SOLEIL,SOMATIQUE,CORPS,SOME,CORPS,SPHERE,SPUERE,
      STAT,ARRET,STASE,ARRET,STATIQUE,ARRET
1540  DATA  STHESIE,SENSIBILITE,STICHE,VERS,STIQUE,LIGNE,STONE,BOUCHE
      ,STROPHE,SILLON,STYLE,COLONNE,SYNCHRONISE,AVEC,SYNTHESE,SYNTHESE
1550  DATA    TAXIE,TOMEAU,TAXIE,ARRANGEMENT,TECHNIQUE,TECHNIQUE,TELIE
      ,LOIN,TERE,FORTER,THEISME,DIEU,THELIA,MAMELON,THEQUE,ARMOIRE,TH
      RAMPSUTHE,QUI SOIGNE
1560  DATA  THLME,CHALEUR,THESE,POSER,TOME,COUPER,TONE,TON,TOPE,LIEU,
      TROP,MACHIN,TROPISME,CROISSANCE,TROPE,QUI TEND VERS,TROPISME,TE
      NDANCE VERS
1570  DATA    TYPE,CARACTERE,UNIVERS,UNIVERS,URIE,URINE,VISION,VISION,
      VORE,QUI DEVORE,XYLE,BOIS,ZOAIRE,ANIMAL,ZOIQUE,ANIMAL,ZTOOTE,ATT
      ELLE,OR,TRUC,FIN,
```

Les exemples d'exécution utilisés à TF1

```
LOGOTRON 1
  UNITE EXPERIMENTALE DE
  LEXICOSYNTHESE
    ACTIVATION DU MODULE SEMANTIQUE
          PARALLELLOSCAPHE
      MOT DE 16 LETTRES
          CODE:SV / 41632.9998
  SIGNIFICATION CALCULEE:
VEHICULE POUR VOYAGER DANS LES UNIVERS
PARALLELLES

LOGOTRON 1
  UNITE EXPERIMENTALE DE
  LEXICOSYNTHESE
    ACTIVATION DU MDDULE SEMANTIQUE
          CLAUSTROCOSME
      MOT DE 13 LETTRES
          CODE:PF / 8524.24982
  SIGNIFICATION CALCULEE:
UNIVERS DU POUSSIN
```

```
LOGOTRON 1
  UNITE EXPERIMENTALE DE
  LEXICOSYNTHESE
    ACTIVATION DU MODULE SEMANTIQUE
          CHRONO THERAPEUTE
      MOT DE 17 LETTRES
          CODE:GP / 7682.74315
  SIGNIFICATION CALCULEE:
MEDECIN QUI SOIGNE EN LAISSANT FAIRE LE
TEMPS

LOGOTRON 1
  UNITE EXPERIMENTALE DE
  LEXICOSYNTHESE
    ACTIVATION DU MODULE SEMANTIQUE
          CRYPTODYNE
      MOT DE 10 LETTRES
          CODE:RW / 42672.4899
  SIGNIFICATION CALCULEE:
VEHICULE DONT LE MODE DE PROPULSION DE-
MEURE CACHE.EX SOUCOUPE VOLANTE
```



tous les gens sont tombés dans le panneau, et des individus fort respectables souhaitaient avoir plus de détails sur la structure de ce fameux « module sémantique. »

Marc concluait : « Nous croyions faire un canular, et nous nous retrouvions avec une science sur les bras : la logotronique. »

Que conclure ? Que la télévision est un instrument puissant. Le petit écran, un stade de « professeur », de « directeur de l'Institut de Logotronique », un ordinateur, et voilà des paroles d'évangile. Nous avons renoncé prudemment au canular numéro 2 : « l'extrapolateur historique » (programme chargé de « calculer le journal du jour à partir de celui de la veille », pour pallier une éventuelle panne des télex des journaux. Nous avions imaginé des rubriques comme : « prévisions politiques pour la journée de demain »).

Nous proposons donc aux lecteurs de L'Ordinateur Individuel de nous aider à constituer, dans un concours permanent, un dictionnaire de logotronique. Bien sûr, vous obtiendrez, pêle-mêle, des mots auxquels vous trouverez difficilement une signification : sonophone ou visioscope, qui seront alors interprétés comme étant des OVNI (c'est-à-dire des objets verbaux non identifiés), plus rarement des mots existants déjà, et quelques trouvailles succulentes. Bonne pêche.

Jean-Pierre Petit
et Marc Arondel

pêche

nous avons trouvé le monstre du Loch Ness !

Outre-Manche, il est difficile de déterminer si le sport national est :
1 Boire du thé
2 Etre membre d'un club
3 Se prendre pour Sherlock Holmes
4 En restant flegmatique

5 Ou aller pêcher le monstre du Loch Ness
Depuis le temps qu'ils le cherchent, ce monstre ! Une véritable artésienne. Nous avons quant à nous résolu le problème et choisi de le laisser résider dans un coin de la mémoire d'un TRS 80. Le trouverez-vous ? Brrr !!!!

```
5  REM  PROGRAMME DE PECHE AU MONSTRE DU LOCH NESS  ***
6  REM  AUTEUR  :  ACKAOUY
7  REM  COPYRIGHT  L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
8  REM
10   PROFONDEUR DU LAC                                  '*
20 DEFINT A-DIM(*,12) FOR X=1TO3 FOR Y=1TO5 READ ZP
30 Z(X Y)=ZP Z(7-X,Y)=ZP Z(X,13-Y)=ZP Z(7-X,13-Y)=ZP
40 NEXT X,Y
50 DATA 3,3,3,3,3,3,5,6,4,6,5,5,5,6,5,5,4,0
60 CLS PRINTGTO(23)PRINT@456,'LE MONSTRE DU LOCH NESS'
70 PRINT PRINT VOULEZ-VOUS LES REGLES DU JEU
80 INPUT QS IF ASC(QS)=79 GOSUB 2000
90 CLS PRINT TAB(13) CARTE DU LOCH'NESS (PROFONDEURS)
100 FOR X=1 TO 6  PRINT TAB(X*9)(  ,  NEXT X  PRINT
102 POST=1TO56 PRINT"-", NEXT
105 FORY=1TO12 PRINTY,(A$(5)  ),
110 FORX=1TO6 PRINTTAB(X*9)Z(X,Y)*100
115 NEXTX PRINT
120 NEXTY
130 INPUT"VOULEZ-VOUS ESSAYER DE PECHER LE MONSTRE ,Y$
140 IF A$(Y$)="O"THEN 150 ELSE 485
150 CLS
160 'POSITION ALEATOIRE DU MONSTRE CLS
170 RANDOM XL=3*(RND(2)-1)+1 XM=XL+RND(3)-1 XM=XM+(RND(5)-1)*.2
180 YL=3*(RND(6)-1)+1 YM=YL+RND(3)-1 YM=YM+(RND(5)-1)*.2
185 XI=XM YI=YM
190 ZM=Z(XM,YM) ZM=RND(ZM) ZI=ZM CLS
200 IF YL=1 THEN YS="NORD-NORD"
205 IF YL=6 THEN YS="CENTRE-NORD
210 IF YL=7 THEN YS="CENTRE-SUD
215 IF YL=12 THEN YS="SUD-SUD"
220 IF XL=1 THEN XS="-OUEST"
225 IF XL=6 THEN XS="-EST"
230 PRINT 'LE MONSTRE A ETE RECEMMENT VU DANS LE SECTEUR  '
    YS  XS
260 PRINT 'DU LAC REPRESENTE CI-DESSOUS  ',
    'COORDONNEES ET PROFONDEUR'
261 FOR X=192 TO 255 STEP192  PRINTTAB(X,STRING$(32,143))  NEXT X
263 PRINT@128,"X",CHR$(130),CHR$(164),"X",
265 FOR Y=128 TO 704 STEP64
267 FOR X=4 TO 28 STEP 8   PRINTTAB(X,STRING$(2,191))  NEXT X
268 NEXT Y
250 AT=64   FOR Y=YL TO YL+2   TA=-1   AT=AT+192
    FOR X=XL TO XL+2   TA=TA+8
255 Z4=Z(X,Y)
270 PRINT@AT,Y, PRINT@128+TA,X PRINT@AT+TA-1,ZA*100,
271 NEXT X  NEXT Y
272 PRINT@896, 'TAPEZ LA BARRE D'ESPACE QUAND VOUS SEREZ PRET.',
274 A$=INKEY$ IF A$<>" "THEN280
276 FOR X=1 TO 500   NEXT  PRINT@46,'                    ',
    FOR X=1 TO 100   NEXT  PRINT@46,V$  W$   GOTO 274
280 PRINT@896,
    INPUT "A QUELLES COORDONNEES LANCEZ-VOUS LE FILET X,Y",XF,YF
285 IF XF<XL OR XF>XL+2 OR YF<YL OR YF>YL+2
    THEN PRINT@96,CHR$(30)  ,  GOTO 280
290 PRINT@896,CHR$(254)CHR$(79),
    INPUT "A QUELLE PROFONDEUR VOULEZ-VOUS LE DEPLOYER",ZF
    ZF=ZF/100
295 ZI=Z(XF,YF) IF ZF>ZI THEN
    PRINT@960,"VOUS N'AVEZ PAS CETTE PROFONDEUR",  GOTO 290
300 PRINT@896,
    "LE FILET EST EN DEPLACE PENDANT LA REMONTEE DU FILET" ;
    TA=(ZF-ZI)*8+(YF-YL)*192+2C2
    FOR Z=ZF TO 1 STEP-1
    PRINT@TA,'FILET ",  PRINT@TA+64,Z*100 ,
305 IF XF=INT(XM) AND YF=INT(YM) AND Z=ZM GOTO 10000
310 FOR T=1TO450 NEXT T PRINT@TA,"
    PRINT@TA,ZI*100, PRINT@TA+65,"      ";FORT=1TO200 NEXT T
315 Z=RND(2)*RND(3)-1
    IF ABS(INT(XM)-XF)=1 AND ABS(INT(YM)-YF)=1
    AND ABS(INT(ZM)-ZF)<=1 THEN 3000
320 Z=RND(3) : Z=INT(AND(3)-2) : ON Z GOTO 330, 340, 350
330 XM=XM+RND.5  IF XM<=1 THEN XM=1.5
335 IF XM>=6 THEN XM=5.5
340 YM=YM+RND.5  IF YM<=1 THEN YM=1.5
345 IF YM>=12 THEN YM=11.5
350 ZM=ZM+D  IF ZM<=0 THEN ZM=1
355 XL=Z(INT(XM),INT(YM)) : IF ZM>ZL THEN ZM=ZL
357 IF XF=INT(XM)AND YF=INT(YM)AND Z=ZM THEN GOTO 10000
360 NEXT Z
    IF ZL<>INT((XM-1)/3)*3+1 OR YL<>INT((YM-1)/3)*3+1 GOTO 370
    ELSE CLS   PRINT@39=0,
    "LE SONAR SIGNALE TOUJOURS LE MONSTRE DANS CE SECTEUR",
    PRINT CHR$(28) ,   GOTO 230
370 PRINT@896,"IL ETAIT EN" ; INT(XI) , INT(YI)
    'PROFONDEUR ", ZI*100 , CHR$(30)
    PRINT@960,"ET S'EST ECHAPPE EN" , INT(XM) , INT(YM) ,
    'PROFONDEUR" , ZM*100 ,
380 PRINT@896, "UN RECOMMENCE?",
390 PRINT@830,"E A T E I ! !",
400 A$=INKEY$  FOR X=1 TO 500   NEXT
    PRINT@830,"                  ",  FOR X=1 TO 200 : NEXT
    PRINT@830,"E A T E I ! ! ",  IF A$=" " THEN 400 ELSE
    IF A$="N" PRINT@575,'AU REVOIR, VOUS PARTIREZ DONC" ,
    PRINT@739,"REEDUILLS.' : GOTO1000
410 GOTO150
445 CLS
450 PRINTCHR$(23) PRINT@274,"L A G H E U R "
    PRINT@524,"Z E D O I L L A R D"+GOTO450
2000 CLS PRINT
     '   LA PECHE AU  M O N S T R E  DU  L O C H ' N E S S "
2010 PRINT PRINT
  "LE LAC EST DIVISE EN 8 SECTEURS REPERES PAR LES DIRECTIONS"
2020 PRINT"GEOGRAPHIQUES: NORD SUD CENTRE EST OUEST"
2030 PRINT
    "LE SECTEUR DU LE MONSTRE A ETE VU RECEMMENT VOUS SERT "
2040 PRINT
 "INDIQUE . VOUS DEVEZ LANCER UN FILET DANS UNE DES REGIONS DE"
2050 PRINT
  "CE SECTEUR ET A UNE PROFONDEUR DE VOTRE CHOIX . LE MONSTRE"
2060 PRINT
     ALORS S'EVEILLE ET SE DEPLACE ALEATOIREMENT DANS LES RIA"
2070 PRINT
  "DIRECTIONS PAR PETITS BONDS . VOUS DAGNEREZ SI LE FILET LE"
2080 PRINT
    'RAMONTE . MONNT PECHE ET ACCROCHEZ VOTRE COEUR DAS IL EST'
2085 PRINT"GLOS , FORT , EPRAYANT ET CRUEL ! ! ",PRINT
2090 INPUT "ETES VOUS PRET " , E$
     IF ASC(E$)=79 THEN RETOUR  ELSE GOTO 2090
3000 FOR L=1TO10
3010 FOR S=1 TO 6   PRINTTAB(S-1,CHR$(164+8*S)) ,  NEXT S
     FOR S=1 TO 6   PRINTTAB(S+63+S,CHR$(132-3*S)) ,  NEXT S
     FOR S=1 TO 6   PRINTTAB(S-1,CHR$(15+8*S)) ,  NEXT S
     FOR S=1 TO 6   PRINTTAB(S+63+S,CHR$(140+7*S)) ,  NEXT S
3015 NEXT L
3020 CLS PRINT@462,"LE MONSTRE A RENVERSE VOTRE RATEAU",
     PRINT@510,"LA GRALOUPE DE SAUVETAGE ARRIVE ! !",
3030 FOR T=1TO2000 NEXT T:IF X<>2 THEN 4000
3040 GOTO 30
4000 CLS   FOR X=10 TO 117 : SET(X,5) : SET(X,62)  NEXT X
     FOR T=5 TO 42   SET(10,T) : SET(11,T) : SET(116,T)  NEXT T
4010 PRINT@272,"  T R O P   T A S O",
     PRINT@340,"LE MONSTRE VOUS A DEVORE",
     PRINT@465,"NOUS PARVIENDRONS VOS PROCHES",
     PRINT@591,"I P P L ",
4020 GOTO4020
10000 CLS
10010 PRINT@75,"BRAVO VOUS AVEZ ATTRAPE  "
10020 FOR T=1TO313
10030 READA
10040 LFX=0THEN10080
10050 READX FORX=ATOX
10060 SET(X,Y)
10070 NEXTX GOTO10030
10080 NEXTY
10090 PRINT@831,"L E   M O N S T R E   D E   L O C H ' N E S S',
10100 FOR T=1TO1000 NEXT
10110 SET(105,12) SET(106,12)
10120 FORX=1TO250 NEXT
10130 RESET(105,12) RESET(106,12)
10140 GOTO10100
10150 DATA33,33,0,26,38,0,24,36,0,23,31,27,25,28,29,28,0,
      23,30,33,46,36,45,100,100,104,106,108,118,0,29,30,45,25,35,
      110,8,29,30,54,24,25,110,0,32,31,53,34,36,97,108,106,107,
      109,0,23,34,67,53,35,26,35,26,100,0,26,53,5+,0
10160 DATA43,28,70,55,30,56,36,100,0,27,33,56,56,10,0,61,28,70,26,
      25,36,35,100,102,102,105,105,0,10,61,46,26,27,24,28,36,35,
      35,100,100,104,106,106,108,0,14,45,51,26,28,33,35,44,32,
      108,8,32,45,51,55,59,62,35,33,35,10,0
10170 DATA28,56,58,42,45,22,25,31,0,10,  5,65,72,36,29,23,28,0,
      41,72,26,29,63,35,0,63,25,0,64,23,0,35,32,36,41,0,28,33,77,
      81,0,68,33,77,81,0,68,35,77,87,0,82,53,78,85,0
30000 GOTO 30000
```


Première parution : L'OI n° 15

pour vos luttes de classes un morpion japonais

Vous souvenez-vous du morpion ? Les Japonais appellent ce jeu Gomoku, mais en France qui n'a pas joué au moins une fois au morpion durant sa scolarité ? C'était bien pratique : il suffisait d'un crayon, d'une feuille de papier quadrillée et d'un adversaire... D'ailleurs, je suis sûr qu'en ce moment même, vous avez tout ce matériel sous la main... sauf peut-être un partenaire ?! Ce n'est pas grave : un petit programme y pourvoiera.

Nous nous sommes désormais fixés un but : écrire un programme qui joue au morpion (1). Soyons ambitieux : il nous faut un programme qui joue « *bien* » et qui nous offre une résistance suffisante. Mais ne nous emballons pas : il faut un programme court et simple. Les articles de David Levy sont passionnants, mais je n'ai pas eu le courage d'écrire un programme complet qui « *voie* » cinq ou six coups à l'avance. D'ailleurs un programme complet réfléchirait sans doute pendant des heures ! Ce serait beaucoup d'énergie dépensée pour un jeu aussi simple. En effet, quoi de plus bête que d'aligner cinq croix ?

« *De plus bête* » ? Voire ! Après un instant d'euphorie, on se rend compte que ce n'est pas si évident. Puisque nous sommes exigeants, nous voulons que ce programme joue toujours une cinquième croix quand quatre sont déjà posées, sinon comment diable arriverait-il à gagner ? Il faut aussi qu'il puisse parer les menaces les plus élémentaires de son adversaire, c'est-à-dire compléter avant lui un alignement de quatre de ses pions. Et puis il faut que le programme sache non seulement exécuter des menaces mais aussi en créer, qu'il évente celles de son adversaire et, pendant que nous y sommes, qu'il sache bien se placer quand rien d'immédiat n'est en vue.

Comment un programme peut-il faire tout cela sans calculer sur plusieurs coups ? Et bien il le fera de la même façon qu'un humain : en regardant le jeu et en déterminant les bonnes cases. Ou comme dit David, nous allons essayer d'avoir une fonction d'évaluation suffisamment bonne pour qu'elle nous permette d'avoir une profondeur d'évaluation très réduite, ici un demi-coup.

Le but du jeu est d'aligner cinq croix (chaque joueur en pose une à son tour, etc.). Il y a exactement 192 façons d'aligner cinq croix sur un quadrillage 10 × 10, c'est-à-dire 192 quintuplets de 5 points alignés.

Le but du jeu est de remplir un (2) de ces quintuplets avec ses propres

(1) Il s'agit d'une variante du morpion jouée sur un espace limité à 10 cases sur 10. Le gomoku se joue sur 19 × 19 cases.

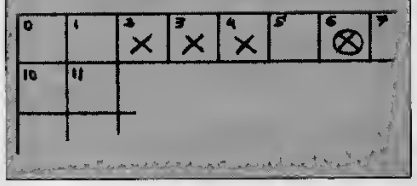
(2) Le problème serait plus complexe dans le cas de la variante du morpion où la partie ne s'arrête pas au premier alignement de 5 mais au contraire continue. Il faudrait alors évaluer à plusieurs coups de profondeur.

Première parution : L'Ordi n° 25

Programme : jeu du morpion

```
10 REM          JEU DU MORPION SUR 10X10 CASES
15 REM          AUTEUR PHILIPPE SCHNOEBELEN
20 REM  COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
25 GOSUB 100                REM   PROLOGUE DU JEU
30 REM  --      BOUCLE DES PARTIES.
35 GOSUB 800                REM   PROLOGUE DE LA PARTIE
40 GOSUB 700                REM   PARTIE
45 PRINT "VOULEZ-VOUS JOUER UNE AUTRE PARTIE ";: GOSUB 9000
50 IF R$= "O" THEN 30
55 GOSUB 900                REM EPILOGUE DU JEU
95 END

100 REM         --      PROLOGUE DU JEU   --
110 PRINT: PRINT  PRINT "MORPION 10X10"
120 DIM A(99),B(99),V(21),B$(5),A$(5)
130 REM --
140 REM LISTE DES TABLEAUX
150 REM         A       : CONTIENT LA SITUATION DU JEU.
160 REM         B       : CONTIENT LES NOTES DES CASES.
170 REM         V       : CONTIENT LES NOTES DES DIFFERENTS
180 REM                   QUINTUPLETS.
190 REM         B$      . CONTIENT 3 VALEURS UTILISEES POUR
200 REM                   L'AFFICHAGE.
210 REM         A$      : LIBELLES DES LIGNES
220 REM --
230
240 REM --      INITIALISATIONS DU JEU
250 REM   NOTES
260 FOR J=0 TO 21
270  V(J)=0  REM    NOTE NULLE POUR QUINTUPLETS NIAIFS...
280 NEXT J
290 FOR J=0 TO 4
300  READ V(J) REM   . . LA QUI N'EST PAS EN CAS DE CEUX-CI. .
310 NEXT J
320 REM          ....   ...0   ..00   ..000   .0000
330 DATA         .01,   .03,    5,    10,    10000
340 FOR J=5 TO 20 STEP 5
350  READ V(J) REM   . . NI DE CEUX-LA.
360 NEXT J
370 REM          ....X   . .XX   .XXX   .XXXX
380 DATA         .1,    2,     100,   1000000
390
400 REM - CARACTERES POUR LES SORTIES
410 READ  B$(0),B$(1),B$(5)
420 DATA   " ", "O", "X"
430 FOR J=0 TO 9
440  READ A$(J)
450 NEXT J
460 DATA  "00 ","10 ","20 ","30 ","40 "
470 DATA  "50 ","60 ","70 ","80 ","90 "
480 REM - POUR LE GENERATEUR DE NOMBRES ALEATOIRES.
490 R=0
500 RETURN

600 REM         --      PROLOGUE DE LA PARTIE   --
610
620 REM - INITIALISATIONS DE LA PARTIE
630 FOR J=0 TO 99
640  A(J)=0: B(J)=0
650 NEXT J
660
670 N=0          REM NOMBRE DE COUPS JOUES
680 F=0          REM INDIQUE QUE LA PARTIE CONTINUE
690 RETURN

700 REM         --      PARTIE  --
710
720 PRINT "VOULEZ-VOUS COMMENCER ";: GOSUB 9000
730 IF R$="O" THEN GOSUB 5000: GOTO 770
740 REM - BOUCLE
750 GOSUB 1000          REM L'ORDINATEUR JOUE.
760 IF F<>0 THEN 800    REM FIN DE LA PARTIE.
770 GOSUB 2000          REM L'ADVERSAIRE JOUE.
780 IF F=0 THEN 740     REM COUP SUIVANT.
790
800 REM - LA PARTIE EST FINIE.
810
820 REM - QUI A GAGNE ?
830 IF   F=1      THEN PRINT "J'AI GAGNE EN",
    ELSE IF F=-1 THEN PRINT "VOUS AVEZ GAGNE EN",
    ELSE          PRINT "PARTIE NULLE APRES",
840 PRINT N;  "COUPS"
850 RETURN

900 REM         --      EPILOGUE DU JEU --
910 PRINT "AU REVOIR"
920 PRINT: PRINT
930 RETURN

1000 REM        --      LE PROGRAMME JOUE       --
1010
1020 REM - REMISE A ZERO DES NOTES -
1030 FOR J=0 TO 99: B(J)=0: NEXT J
1040
1050 PRINT "JE JOUE ";
1060 REM - APPELS POUR LES QUATRE DIRECTIONS EN PREMANT
1070 REM   CARDE AUX LIMITES.
1080
1090 C=1: J1=0: J2=5: K1=0: K2=90: GOSUB 1500
1100 C=9: J1=4: J2=9: K1=0: K2=50: GOSUB 1500
1110 C=10: J1=0: J2=9: K1=0: K2=50: GOSUB 1500
1120 C=11: J1=0: J2=5: K1=0: K2=50: GOSUB 1500
1130
1140 REM TOUTES LES CASES SONT MAINTENANT NOTEES
1150
1160 REM ON VA DETERMINER "LA" MEILLEURE, OU L'UNE DES MEILLEURES.
1170 S=-1          REM MEILLEUR SCORE JUSQUE-LA
1180 Q=0           REM NOMBRE DE CASES AYANT LE SCORE S.
1190 REM           C   CASE CHOISIE.
1200 FOR J=0 TO 99
1210  IF A(J)<>0 OR B(J)<S THEN 1230
1220  IF B(J)>S THEN C=J: S=B(C): Q=1
                 ELSE Q=Q+1: IF Q*RND(R)<1 THEN C=J
1230 NEXT J
1240 PRINT: PRINT C
1250 A(C)=5
1260 GOSUB 4000          REM AFFICHAGE
1270 N=N+1               REM NOMBRE DE COUPS JOUES
1280 IF N=100 OR (S>0 AND N>1) THEN F=2
1290 IF S>=V(20) THEN F=1        REM LA PROGRAMME A GAGNE
1300 RETURN

1500 REM        -       EXPLORATION DES QUINTUPLETS     -
1510 REM CETTE EXPLORATION SE FAIT DANS LA DIRECTION C
1520 C=-A*C
1530 FOR J=J1 TO J2
1540  FOR Z=K1 TO K2 STEP 10
1550  A=J+Z: B=A+C4: Z=0
1560  REM - NOTATION DU QUINTUPLET.
1570  FOR Z=A TO B STEP C: Z=Z+A(Z): NEXT Z
1580  Q=V(Z)      REM ON NOTE LA CASE
1590  IF Q<>0 THEN FOR Z=A TO B STEP C: B(Z)=B(Z)+Q: NEXT Z
1600 NEXT K
1610 NEXT J
1620 RETURN

2000 REM        --      L'ADVERSAIRE JOUE       --
2010 INPUT "VOTRE COUP ";A
2020 REM - COUP CORRECT ?
2030 IF A<0 OR A>99 OR A<>INT(A) THEN
     PRINT "UN ENTIER ENTRE 0 ET 99 SVP...";: GOTO 2010
2040 IF A(A)<>0 THEN PRINT "CASE DEJA OCCUPEE ";: GOTO 2010
2050
2060 A(A)=1              REM COUP CORRECT.
2070 GOSUB 4000          REM AFFICHAGE.
2080 N=N+1               REM NOMBRE DE COUPS JOUES
2090
2100 IF N=100 THEN F=2          REM PARTIE NULLE.
2110 IF B(A)>=V(4) THEN F=-1    REM IL A GAGNE
2120 REM ****** ATTENTION !
2130 REM *      L'INSTRUCTION CI-DESSUS NE MARCHE QUE PARCE QUE
2140 REM *      - LA VALEUR DE B() EST MISE A JOUR A CHAQUE COUP
2150 REM *        DE L'ORDINATEUR.
2160 REM *      - SI LE JOUEUR COMMENCE, B(A) VAUDRA 0, QUI EST
2170 REM *        UNE VALEUR CORRECTE.
2180 REM *      - SI B(A)>=V(4), L'UN DES DEUX ADVERSAIRES A
2190 REM *        FORCEMENT GAGNE; CE NE PEUT ETRE LE PROGRAMME,
2200 REM *        PUISQUE LA PARTIE A CONTINUE APRES SON COUP.
2210
2220 RETURN

4000 REM        --      AFFICHAGE       --
4010 PRINT: PRINT "   0 1 2 3 4 5 6 7 8 9"
4020 FOR J=0 TO 9
4030  PRINT A$(J);
4040  FOR K=0 TO 9
4050   PRINT B$(A(10*J+K));
4060  NEXT K
4070  PRINT
4080 NEXT J
4090 RETURN

9000 REM        --      OUI OU NON ?    --
9010 INPUT R$: R$=LEFT$(R$,1)
9020 IF R$<> "O" AND R$<> "N" THEN PRINT "OUI OU NON ";: GOTO 9010
9030 RETURN
```

Figure 1



croix et ceci *dans n'importe quel ordre*, ce dernier point étant très important

Le programme va donc examiner chacun de ces 192 quintuplets et, suivant ce qu'ils contiennent, les cases restées libres se verront attribuer une bonne ou une mauvaise note, puis l'ordinateur jouera sur la mieux notée. Puisque plusieurs quintuplets différents contiennent une même case, celle-ci se verra attribuer la somme des notes reçues vis-à-vis de chacun des quintuplets.

Par exemple *si un quintuplet est formé de quatre croix* (l'ordinateur a les croix, et le joueur les ronds), il est très intéressant de jouer sur la cinquième case : nous lui attribuerons *une note de 1000000* (!) pour encourager l'ordinateur à s'y placer. *Si par contre il contient quatre ronds* (les nôtres) il faut que l'ordinateur nous empêche de nous y mettre en s'y posant avant nous. Ce sera donc une bonne note, mais elle ne doit pas être meilleure que la précédente car sinon l'ordinateur pourrait oublier de gagner pour (bêtement !) parer une de nos menaces : *adjugeons-lui 10000 « seulement »*.

Si un quintuplet contient au moins une croix et un rond, aucun des adversaires n'a le moindre espoir de s'y installer victorieusement un jour, une note de 0 en est le juste châtiment. Il n'y a donc qu'à examiner les cas suivants :

3 croix et 2 cases vides, c'est très bon d'y jouer car cela crée une menace, *la note sera de 100*.

3 ronds et 2 cases vides, il faut empêcher l'adversaire de nous menacer mais il ne faut pas se détourner d'objectifs plus importants : *une note de 10*.

et ainsi de suite en descendant.

C'est un système très pratique : l'ordinateur représente dans sa mémoire une case vide par un 0, un rond par un 1 et une croix par un 5. Il lui suffit d'additionner les cinq éléments du quintuplet pour connaître son contenu à une permutation des éléments près, mais nous avons dit que cela n'avait aucune importance (3). Une table lui donne alors la note correspondant à cet arrangement, il l'ajoute à un tableau qui contiendra la note de chaque case quand tous les quintuplets auront été examinés. Il paraît y avoir un défaut dans cette position (voir figure 1 page précédente). Il est clair qu'en jouant en 1, l'ordinateur crée une double menace imparable et gagne, mais avec notre système, en examinant le quintuplet (0, 1, 2, 3, 4), l'ordinateur attribuera la même note de 10000 aux cases 0 et 1, or elles ne sont absolument pas équivalentes car en jouant en 0, on permet à l'adversaire de s'en tirer en jouant en 1.

Regardons ceci de plus près : la case 0 reçoit bien la note de 10000, mais la case 1 reçoit elle 20000 points car non seulement le quintuplet (0, 1, 2, 3, 4) mais aussi (1, 2, 3, 4, 5) lui attribuent 10000 points. En fin de compte l'ordinateur jouera toujours en 1, et gagnera.

C'est par ce principe que l'ordinateur parvient à se placer à l'intersection des alignements intéressants. Ce système fait du programme un adversaire très coriace, pas invincible, mais difficilement vaincu. De plus il est très simple à mettre en œuvre, voyez, le programme est très court, la partie qui « réfléchit » n'occupe que 25 lignes.

(3) Notez que le contenu 5 avec 5 ronds ne peut se produire puisque alors le programme aurait perdu, et qu'il aurait signalé ce fait : il n'a donc pas à jouer. Les inquiets pourraient mettre 6 pour chaque croix, à condition de changer le programme en conséquence (tableau V).

Quelques commentaires sur le programme.

De 10 à 95, on a affaire au programme principal selon une architecture classique.

De 100 à 500 la phase d'initialisation. Les possesseurs d'un TRS 80 doivent ajouter en 115 POKE 16553,255 et peuvent y mettre également DEFINT A, C-P, W-Z pour améliorer les performances.

De 600 à 690 l'initialisation de la partie (pour qu'on puisse en jouer plusieurs d'affilée).

De 700 à 850 la partie proprement dite : on détermine qui joue en premier (720-730), et on va en *800* quand la partie est terminée. L'affichage du résultat de la partie se fait en 830, suivant la valeur de F (1, -1 ou 2).

De 900 à 930 l'épilogue du jeu. On pourrait rappeler le nombre de parties nulles et gagnées par chacun, mais ceci risque d'irriter les joueurs trop souvent battus par le programme.

A partir de 1000, le jeu du programme. On explore les 192 quintuplets suivant les 4 directions possibles en 1090-1120, puis on choisit la meilleure case. La ligne 1220 permet de choisir au hasard entre plusieurs cases ayant la même note, mais on peut la supprimer si l'on n'a pas de fonction RND ou si l'on souhaite accélérer le programme.

La partie cruciale du programme est *le sous programme 1500*. L'intérieur de la boucle est effectué 192 fois par coup, et les deux boucles sur P le sont 5 fois plus. Amateurs de langage machine, c'est ce sous-programme qu'il vous faut attaquer, attention car si A et Z sont entiers, V, Z et Q ne le sont pas.

L'adversaire joue en 2000 à 2220, son coup est bien entendu vérifié.

Le sous programme 9000 est le sous-programme OUI ou NON classique.

Le sous-programme 8000 assure l'affichage, chacun l'adaptera à son ordinateur pour que ce soit joli !

Par exemple, pour le rendre plus agréable que ce continuel déroulement, le jeu doit rester fixe et seule la case concernée doit se transformer. Il faut aussi écarter les pions pour y voir un peu plus clair, mais ceci demande un travail propre à chaque matériel utilisé et nous laissons le soin aux utilisateurs de se débrouiller avec leur engin.

Une idée pour s'approcher de la perfection serait d'écrire un deuxième programme en changeant les valeurs des lignes 330 et 380, puis d'opposer les deux programmes et enfin de ne retenir que les valeurs du vainqueur en les considérant comme plus proches de la réalité. Si vous faites des découvertes à ce propos, communiquez-les ! Faites-nous aussi part des valeurs qui permettent à la machine de « cafouiller » le plus lamentablement, et donc à l'homme de gagner à tout coup : on peut ainsi avoir un jeu à difficulté variable.

Et si vous désirez ensuite rendre votre jeu plus intéressant, plus captivant et plus difficile, lancez-vous à l'attaque du Gomoku japonais qui se joue sur un damier de jeu de go, soit un univers de 19 × 19 = 361 intersections et, petit détail, il est interdit de créer un ensemble de pions contenant 2 lignes de 3 pions libres (ronds ou croix), ensemble qui permet de gagner dans la plupart des cas au morpion français.

Pour les fanatiques, le gomoku possède lui-même des variantes, dont la variante ninuki où l'on retire du jeu des pions posés : elle a été présentée par notre confrère « Jeux et Stratégie » dans son numéro 1.

Une autre variante consiste à remplir cet univers 19 × 19 dans son entier ce qui créera plusieurs « morpions » pour chaque joueur, celui qui en a le plus étant déclaré gagnant. Dans ce cas, deux options sont possibles : soit on ne rejoue pas, soit on rejoue après avoir effectué un « morpion ».

Mais avant ces développements vous avez les ronds, votre machine les croix ! A vous de jouer !

Philippe Schnoebelen

à moi compte, deux mots !

Qui ne s'est jamais assis, volontairement ou par hasard, un soir vers 19 heures devant un poste de télévision présentant l'émission « des chiffres et des lettres » ? Peu de gens sans doute. Tous ceux qui ont essayé de retrouver « le mot le plus long » seront donc déçus par le programme que nous présentons car nous n'avons pas encore fini de mémoriser le dictionnaire. En revanche, ce programme leur permettra de lancer bien souvent la phrase « le compte est bon ».

Rappelons tout d'abord le problème. On fournit 6 nombres parmi 1, 2, ... 10, 25, 50, 75, 100 et un nombre de 3 chiffres. Le joueur, ici l'ordinateur, doit reconstituer ce dernier nombre, par des opérations simples sur les 6 premiers nombres.

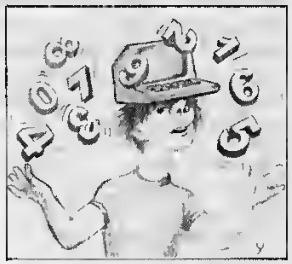
Le programme BASIC n'aboutit que lorsqu'il trouve le compte exact, ce qui heureusement lui arrive dans la plupart des cas. Le principe est assez simple et montrons le sur un exemple : soit à trouver 723, avec les nombres 4, 5, 6, 7, 8, 9.

Le programme essaie successivement (723-4)/5, (723+4)/5, (723—4)/6 ... jusqu'à trouver un nombre entier : ici, l'opération s'achève pour (723+5)/7= 104. Et la méthode reprend : (104—0)/8= 13, (13—4)/9= 1. Ici le processus s'achève, et on peut reconstituer 723 dans l'ordre inverse : 9 × 1= 9, 9+4= 13, 13×8= 104, 104+0= 104, 104×7= 728, 728—5= 723.

L'affichage supprime les lignes inutiles :

Le compte est bon
en effet
9 + 4 = 13
13 × 8 = 104
104 × 7 = 728
728 — 5 = 723

Le programme tel qu'il est conçu ne permet pas de faire plusieurs additions successives, mais ce n'est dans la pratique qu'un petit inconvénient, et il trouve au moins 95 % des comptes possibles. Initialement conçu sur Pet, il peut fonctionner avec le plus pauvre des BASIC. Sur Pet, ce programme conclue le plus souvent dans les 45 secondes, mais il est possible qu'avec certains BASIC trop lents, le temps de réponse dépasse parfois la minute (il lui arrive de trouver aussi en quelques secondes). Dans tous les cas, ce programme est réellement très efficace et beaucoup de candidats du jeu télévisé pourraient renverser leur situation, si un ordinateur individuel pouvait tenir dans une poche...



Les fanatiques verront vite que parfois des comptes évidents échappent à la logique du programme. Il ne leur reste donc plus qu'à optimiser celui-ci pour disposer du temps nécessaire à l'ajout des éléments logiques qui permettront de retrouver la solution dans tous les cas où il en existe une.

Jean Claude Buisson

```
100 rem                    le compte est bon
110 rem                    auteur : j.c. buisson
120 rem      copyright l'ordinateur individuel et l'auteur
130 rem ---- ----- ----------------------------------------------------------
140 print "les 6 nombres avec"
150 input n(1),n(2),n(3),n(4),n(5),n(6)
160 input "quel est le nombre à trouver";x : g(0)=x : a(0)=1 : a(3)=1
170 for i = 1 to 6 : g(i)=n(i) : next i
180 rem ----------------------------------------------------------------------
190 ann=1
200 i(a)=i(a)+1 : if s(i(a))=0.1 then 270
210 j(a)=j(a)+1 : if j(a)=7 then 210
220 if s(j(a))=0.1 or j(a)=i(a) then 250
230 a(a)=(a(a-1)*n(i(a)))/n(i(a)) : if a(a)=abs(int(a(a))) then 310
240 g(a)=(g(a-1)*n(i(a)))/n(j(a)) : if g(a)=abs(int(a(a))) then 320
250 if j(a) 6 then 210
260 i(a) 0
270 if s(a) 7 then 200
280 i(a) 0
290 if a=1 then 500
300 a=a-1 : g(i(a))=n(i(a)) : s(j(a))=n(j(a)) : goto 250
310 s(a)=1 : goto 530
320 s(a)=1
330 if a(a)=0 or a(a)=1 then 410
360 if i(a)=7 then 360
350 s(i(a))=0 t
360 if j(a)=6 then 190
370 s(j(a))=0 t : goto 190
380 rem ----------------------------------------------------------------------
390 rem                    affichage du resultat
400 rem
410 print : print : print g(0) ; "le compte est bon"
420 prints : print tab(13) "en effet" : print
430 for k=0 to 1 step -1
440 if a(k) 1 or n(j(k))=1 then 470
450 if g(k)=1 or n(i(k))=0 then 480
460 print tab(10);n(i(k)); " " ;s(i(k)); " " ;n(j(k)) : if n(i(k))=0 then 480
470 print tab(10);g(k);s(k);n(j(k)); " " ;g(k-1)
480 next k
490 print : print tab(15) , "bof" : goto 510
500 print : print tab(10) , "je n'y arrive pas
510 end
```

l'angoisse de l'ordinateur au moment du penalty

Avec son nom bien de chez nous, le football est depuis longtemps le sport national français. Les passionnés du ballon rond, qu'ils fréquentent les terrains ou les tribunes (les fauteuils de TV ?), se recrutent aussi parmi les « péessistes ». Quoi de plus déprimant alors que d'être à la merci des erreurs des chroniqueurs sportifs quant à l'établissement du classement. Il est temps de prendre notre revanche ! Le carnet du supporter en BASIC... ou : l'OI au service du ballon rond.

Rappelons d'abord le règlement du championnat de France 1re Division. Il y a N = 20 équipes en lice, chaque équipe rencontre deux fois chacune des N — 1 = 19 autres, une fois à domicile, une fois à l'extérieur. Par conséquent, le championnat est divisé en (N — 1) × 2 = N × 2 — 2 = 38 journées comportant chacune N/2 = 10 matches. Un match gagné rapporte 2 points, un match nul rapporte 1 point, un match perdu ne rapporte rien. Il y a donc 2 points en jeu par match, enfin, le classement est fonction du nombre de points de chaque équipe. L'équipe ayant le plus de points est première, l'équipe en ayant le moins est dernière.

En outre, si deux équipes ont le même nombre de points, on les départage en fonction de leur différence de buts : buts marqués — buts encaissés. L'équipe dont la différence de buts est la plus élevée est classée avant l'autre.

En cas de nouvelle égalité à la différence de buts, c'est l'équipe ayant la meilleure attaque, c'est-à-dire celle qui a marqué le plus de buts, qui précède l'autre au classement.

Si encore deux équipes sont à égalité, elles sont déclarées ex aequo (très improbable). De plus, l'équipe championne (celle qui occupe la première place à l'issue de la dernière journée) participe l'année suivante à la Coupe d'Europe des clubs champions, tandis que ses deux (ou serait-ce trois maintenant ?) dauphines participent à la Coupe d'Europe de l'U.E.F.A.

Les deux dernières sont reléguées en 2e Division, remplacées par les premières de chaque groupe de 2e Division. L'antépénultième doit disputer les matches de barrage avec le vainqueur des 2e de chaque groupe de 2e Division, pour défendre sa place en 1re Division.

Le programme assure le traitement des résultats de tous les matches, et donne le classement à l'issue de n'importe quelle journée. Il permet l'examen et la modification de toutes les données.

Ce programme a été écrit sur Apple II avec disque, en BASIC Microsoft (Applesoft). Seules seront à changer pour d'autres systèmes les routines d'entrée/sortie (sur disque ou sur cassette). Une instruction fréquemment utilisée peut se nommer différemment sur d'autres OI. HOME efface l'écran et renvoie le curseur en haut et à gauche.

Les variables

Variables constantes

N : lue en DATA, le nombre d'équipes en lice,

EQ $ (N) : tableau des noms de chaque équipe, lus en DATA.

JO $ (N × 2 — 2) : tableau des dates des différentes journées, lues sur disque.

LI $: contient 30 « . » pour la mise en page.

Variables générales

MA (N,N) est un tableau de réels dans lequel sont stockés les résultats de tous les matches.

MA (L,J) = — R si le match n'a pas encore été joué (R est la journée à laquelle appartient le match),

MA (L,J) = + BL.BJ,R, c'est-à-dire BL × 100 + BJ + R/100 avec BL le nombre de buts marqués par l'équipe L et BJ le nombre de buts marqués par l'équipe J. On a donc deux chiffres par élément : BL, BJ et R,

. MA (L,J) = 0 s'il n'y a pas de match (si L = J ou si L ou J est nul)

CL % (N,6) est un tableau d'entiers (comme l'indique le signe %) contenant les différents éléments du classement :

. CL % (L,0) contient l'indice dans EQ $ de l'équipe classée L ième.

. CL % (L,1) contient son nombre de points,

. CL % (L,2) contient le nombre de matches qu'elle a remportés,

. CL % (L,3) contient le nombre de matches où elle a fait nul,

. CL % (L,4) contient le nombre de matches qu'elle a perdus,

. CL % (L,5) contient le nombre de buts qu'elle a marqués,

. CL % (L,6) contient le nombre de buts qu'elle a encaissés.

Autres variables

L,J,K	variables de boucles
L	aussi paramètre de la routine 1100
BL,BJ	buts de l'équipe L et J
	aussi indice de boucles
R	valeur d'une réponse
R,R1	numéro des journées sur lesquelles s'effectue le traitement (R1 pour comparaison avec R)
R$	réponse : équipe, ou date, ou chiffre
R$,R1$	nom des équipes entrées au clavier pour un match
AA	année, MM : mois, JJ : jours, MAX : nombre de jours dans MM (routine 1100)
D$	variable spécifique à l'APPLE, employée dans les entrées/sorties sur disque

Examinons maintenant *les sous-programmes*.

Tout le programme est organisé en sous-programmes plus ou moins imbriqués, qui peuvent être de deux types :

- le type procédure, qui ne produit pas de résultat mais exerce une action sur les données ou les périphériques ;
- le type fonction qui, en plus d'exercer cette action (éventuellement), produit un résultat (numérique ou alphanumérique).

Chacun de ces deux types peut demander le passage d'un ou plusieurs paramètres qui seront soit la « cible » du sous-programme (passer L en paramètre provoquera une action sur l'élément L d'un tableau), soit des critères d'action (journée R pour le classement). Ces conceptions, directement issues de PASCAL, ne sont utilisables que de façon très primitive en BASIC.

Il s'agissait des résultats de la journée du 22 août 1980 bien sûr.

Voici une description de tous les sous-programmes dont on remarquera qu'ils sont présentés dans l'ordre inverse de leur appel pour des raisons d'optimisation.

1) OUI ou NON
numéro : 1000

2) choix ou menu
numéro : 1600

3) entrée MA (N,N)
numéro : 2300

4) entrée JO $ (N × 2 — 2)
numéro : 2700

Ce programme ne s'applique évidemment pas qu'au championnat de France de football de 1re Division. En changeant les DATAs (nombre et noms des équipes), il s'applique à tout tournoi appliquant le même type de règle. Pour un tournoi d'échecs, par exemple, il suffira de diviser le nombre de points par deux.

5) sortie de MA (N,N)
numéro : 2500

6) sortie de JO $ (N × 2 — 2)
numéro : 2800

7) entrée de la journée
numéro : 1200

8) entrée du résultat match (L,J)
numéro : 3200

9) affichage « journée », R.
numéro : 2900

10) affiche une ligne de match
numéro : 3000

11) affiche une ligne de résultats
numéro : 3100

12) liste des dates
numéros : 1400

13) liste des équipes
numéro : 1500

14) vérifie la date
numéro : 1100

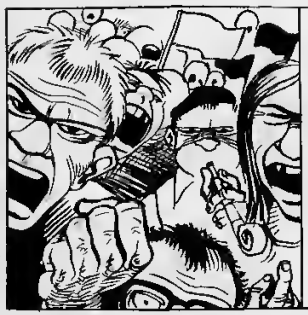
15) classement
numéro : 800

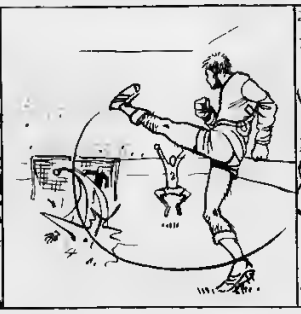
16) initialisation du classement
numéro : 700

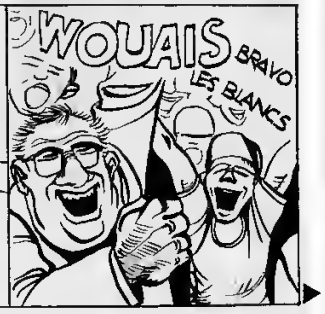
17) traitement des résultats
numéro : 300

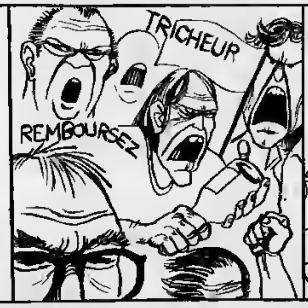
18) tri « bulle »
numéro : 600

19) inversion
numéro : 500









Première parution : L'OI n° 26

20) initialisation
numero 1700
paramètres —
résultat —
appelle 1000 (oui ou non)
2300 [input MA (N,N)]
2700 [input JO $ (N × 2 — 2)]
1900 (1re initialisation) par GOTO
appelé par O (programme principal)

Ce sous-programme important initialise toutes les données. En particulier, il effectue un GOTO 1900 en cas de 1re utilisation du programme

21) 1re initialisation
numéro 1900
parametres —
resultat —
appelle 1100 (vérifie la date)
1500 (liste des equipes)
1200 (input journée)
appele par O par l'intermédiaire de 1700
variables L, J, K boucles
R $, R, R 1 $, R 1 équipes et leurs numeros
BL, BJ boucles

Dans ce sous-programme, on rentre le calendrier, épreuve assez fastidieuse qui necessite l'entrée au clavier de 380 matches. Autant dire que, au début au moins, le nom des équipes en DATA a intérêt a être bref. Il met en drapeau MA (0,0) a 1 de façon à ce que 10000 (fin) sauve le tableau. Il sauve egalement JO $ (N × 2 — 2). Il comporte aussi une routine de vérification d'erreur (2120 à 2200) mais elle est très élementaire puisqu'elle verifie que chaque match appartient à une journée, et seulement cela. Elle ne verifie pas qu'une equipe joue deux matches lors de la même journée par exemple. Les améliorations ici sont nombreuses.

22) entrée des resultats
numéro 5000
paramètres —
resultat —
appelle 1000 (oui ou non)
2900 (print « journée », R)
1200 (input journée)
3100 (affiche une ligne de resultats)
3200 (input résultat match L, J)
1500 (liste des équipes)
appelé par O (programme principal)
variables R $, R 1 $ équipe, L, J leur numéro
K boucle

Ce sous-programme parcourt tous les matches et en demande le résultat s'ils appartiennent à la journée choisie, en informant au préalable l'utilisateur si ce match a déjà un résultat (si on rentre les resultats par journée), on demande de la même façon le résultat du match choisi (si on rentre les résultats par match).

23) affichage des résultats d'une journée
numéro 6000
paramètres —
résultat —
appelle 1200 (input journée)
2900 (print « journée », R)
3100 (affiche une ligne de résultats)
appelé par O (programme principal)
variables L, J boucles

24) résultat d'un match
numero 7000
parametres —
résultat —
appelle 1500, 2900, 3100, 1000
appele par O

C'est le même sous-programme que le précedent, mais sur un match seulement. En 7900 se trouve un point d'entrée spécial de la routine 8000 (affichage du classement), accédé par les trois précédentes, et qui donne le classement à l'issue de la journée R.

25) affichage du classement
numéro 8000
parametres — (R journee si accès en 7900)
résultat —
appelle 800 (classement)
1000 (oui ou non)
appelé par O (5000, 6000 et 7000 à travers 7900)
variables R journée, L, J boucles
R $ des blancs
K le nombre de matches joues

Ce sous-programme effectue l'affichage et la mise en page du tableau de classement. Il affiche une colonne de plus que celles contenues dans CL %: ce sont les matches joués dont le nombre est égal, pour chaque équipe, à la somme des matches gagnés, nuls et perdus. Le classement est établi pour la journée R si on utilise le point d'entrée 7900, et pour la dernière journée si on entre en 8000. Cette routine ne calcule le classement que s'il n'est pas déjà contenu dans CL %. On peut ainsi gagner la minute d'attente quand elle n'est pas necessaire. Cela arrive lorsque la journée demandée est égale à celle contenue dans CL % ou quand CL % contient le dernier classement et que la journee demandée est postérieure.

26) fin
numero 10000
parametres MA (0,0)
appelle 2500 [output MA (N,N)]
appele par O (programme principal)
variables —

Liste du programme

```
1  REM ********************
2  REM PIC LEA ** AS SUPPORTER *
3  REM P      LEA EA-4      *
4  REM FAUTEUR DEA FOOTBALL DUTS *
5  REM *                    *
6  REM *L DAUPHINOIS 1983 JUNALE*
8  REM ********************
100  PRINT " MENU"
110  GOSUB 1700  REM INITIALISATION
120  REM MENU
130  CLS
140  PRINT  LEA( 5) "** LE CARNET DU SUPPORTEUR **"
150  PRINT:  PRINT:  PRINT
160  PRINT " 1) ENTREZ LES DERNIERS RESULTATS "
170  PRINT "2) VOIR LES RESULTATS D'UNE JOURNEE "
180  PRINT "3) AFFIC LE RESULTAT D UN MATCH "
190  PRINT "4) VOIR LE CLASSEMENT"
200  PRINT "5) CALENDRIER D'UNE EQUIPE.
205  PRINT "6) FIN"
210  PRINT
220  PRINT  LEA( 10), "ETTRE CHOIX 1"
230  GOSUB 1000
240  ON R GOSUB 5000,7000,800,1100,10000
250  GOTO 100
297  REM TRAITE LES DONNEES D UN MATCH
298  REM MATCH(L,J)
299  REM ZIGZAG E CENTRE EQUIPE J
300 R1 = PN DEC(MA(L,J)) - INT (MA(L,J))) * 100)
317  IF R1 > 0 THEN 400
320 R4 = PN DEC( INT (MA(L,J) / 100))
330 CLE(L,J) = CLE(L,J) + (RL = RJ)
340 CLE(J,L) = CLE(J,L) + (RJ = RL)
350 CLE(L,2) = CLE(L,2) + (RL > RJ)
370 CLE(J,2) = CLE(J,2) + (RJ > RL)
380 CLE(L,3) = CLE(L,3) + (RL < RJ)
390 CLE(J,3) = CLE(J,3) + (RJ < RL)
400 CLE(L,5) = CLE(L,5) + RL
410 CLE(J,5) = CLE(J,5) + RJ
420 CLE(L,6) = CLE(L,6) + RJ
430 CLE(J,6) = CLE(J,6) + RL
440 CLE(L,4) = CLE(L,4) + (RL > = RJ) + (RL > RJ)
450 CLE(J,4) = CLE(J,4) + (RJ > = RL) + (RJ > RL)
460  RETURN
499  REM INVERSION DES ELEMENTS DE CLE
500  FOR L = 1 TO N
510 R2 = CLE(J,K)=CLE(J,K) = CLE(J,K CLE(J + 1,K) = R1
520  NEXT K
530  RETURN
599  REM AJOUTER UN PAS DE CLE(L,5)
600  REM SELON CLE(L,5) POUR CLE( 5)=CLE(J,6),POUR RELAN CLEC( 5)
690  FOR L = N  1 TO 1 STEP - 1
700  FOR J = 1 TO L
710  IF CLE(J 1) < > CLE(J + 1,1) THEN 550
720  IF CLE(J 5)  CLE(J 5) = CLE(J + 1,5)  CLE(J + 1,5) THEN  IF CLE(J,5) < CLE(J + 1,5) THEN  GOSUB 500
740  IF CLE(J,5) < CLE(J 5) < CLE(J + 1,5) = CLE(J + 1,5) THEN  GOSUB 500
750  IF CLE(J,1) < CLE(J + 1 1) THEN  GOSUB 500
760  NEXT J
770  NEXT L
780  RETURN
799  REM EFFACEMENT DU CLASSEMENT
790  REM VIDE A MOITIE DE LA JOURNEE
800  PRINT  ATTENDRE UN MINUTE
810  GOSUB 700
820  FOR L = 1 TO N
830  FOR J = 1 TO N
840  IF MA(L J) > 0 THEN  GOSUB 930
860  NEXT J
870  NEXT L
880  GOSUB 600
890 CLE(0 5) = R: IF R > CLE(0 5) THEN CLE(0 5) = CLE(0 -)
900  RETURN
929  REM ZIZ DU MOT
1000  GET R$: IF R$ = "" THEN 1020
1010  IF R$ < > "N" THEN 1030
1020  PRINT R$
1030  RETURN
1039  REM VERIFIE LA DAT.
1050  REM VAR JOS(L)
1060  REC PORLAS JJJ MMAAA
1070 AA = VAL ( MID$(T$ (JOS(L),4))
1080 JJ =  VAL (JOS(L))
1120 MM =  VAL ( MID$ (JOS(L,3 5 ( ELSE (J"*(L,5 1) = "/"))
1130  IF MM < 1 OR MM > 12 THEN JOS(L) = "?"  RETURN
1150 MA = JJ + ((MM < 8 AND MM MOD 2 = 1) OR (MM  > MM 2 AND  INT (MM / 2) = MM /2))
1160  IF MM = 2 THEN MA = JJ + (( INT (AA / 4) = AA / 4) AND ( INT (AA / 100) < > AA / 100 OR  INT (AA / 400) = AA / 400))
1170  IF JJ > MA OR JJ < 1 THEN JOS(L) = "?
1180  RETURN
1199  REM INPUT JOURNEE
1200  INPUT  QUELLE JOURNEE ? R$
1210  IF R$(1) = R$(2) = R$  GOSUB 1100
1230  IF JO$(R) = "" THEN 1250
1240  FOR J = 1 TO N * 2  2
1260  IF JO$(J) = R$ THEN R$ = STR$ (J):J = N * 2  N JO$(L) = "
1270  NEXT J
1280  IF JO$(L) < > " THEN R$ = "
1290 K = VAL (R$)
1299  IF K < 1 OR K > N * 2  2  THEN  PRINT "ERREUR !": GOSUB 1400  GOTO 1200
1300  RETURN
1399  REC  LISTE DES DATES
1400  PRINT  VOULEZ VOUS UNE LISTE DES DATES ?",
1410  GOSUB 1000  IF R$ = "N" THEN 1440
1420  FOR J = 1 TO N * N - 2
1430  PRINT J JOS(J)
1440  IF J / 22 < > INT (J / 22) THEN 1470
1450  REM ON S'ARRETE TOUS LES 21 DATES, SELON CAPACITE DE L'ECRAN
1460  PRINT  ON CONTINUE ?  GOSUB 1000
1470  IF R1 = "" THEN J = N * 2 - 2
1480  NEXT J
1490  RETURN
1499  REM LISTE DES EQUIPES
1500  PRINT "VOULEZ-VOUS LA LISTE DES EQUIPES ?"
1510  GOSUB 1000: IF R$ = "N" THEN 1520
1530  FOR  = 1 TO N
1540  PRINT EQ$(R)
1550  NEXT R
1559  REM ENTREE DU CLOTE AU BAND
1560  REM ON ENTRE 1 MATCH : TOT 1 = T 3
1570  = 1 K
1580 L = INT (K/N)
1590 J = K - L (N)
1600  IF L < 1 OR J < 1 OR J > N THEN 1620
1630  RETURN
1699  REM INPUT JOURNEE
1700  PRINT "QUELLE JOURNEE ? R$
1710  IF R$(1) = R$(2) = R$  GOSUB 1100
1720  IF JO$(R) = "" THEN 1750
1730  FOR J = 1 TO N * 2  2
1740  IF JO$(J) = R$ THEN R$ = STR$ (J):J = N * 2  N JO$(L) = "
1750  NEXT J
1760  IF JO$(L) < > " THEN R$ = "
1770 K = VAL (R$)
1779  IF K < 1 OR K > N * 2  2  THEN  PRINT "ERREUR !": GOSUB 1400  GOTO 1750
1790  RETURN
1799  REC  LISTE DES DATES
1800  PRINT  VOULEZ VOUS UNE LISTE DES DATES ?",
1810  GOSUB 1000  IF R$ = "N" THEN 1840
1820  FOR J = 1 TO N * N - 2
1830  PRINT J JOS(J)
1840  IF J / 22 < > INT (J / 22) THEN 1870
1850  REM ON S'ARRETE TOUS LES 21 DATES, SELON CAPACITE DE L'ECRAN
1860  PRINT  ON CONTINUE ?  GOSUB 1000
1870  IF R1 = "" THEN J = N * 2 - 2
1880  NEXT J
1890  RETURN
1899  REM LISTE DES EQUIPES
1900  PRINT "VOULEZ-VOUS LA LISTE DES EQUIPES ?"
1910  GOSUB 1000: IF R$ = "N" THEN 1920
1930  FOR  = 1 TO N
1940  PRINT EQ$(R)
1950  NEXT R
1959  REM ENTREE DU CLOTE AU BAND
1960  REM ON ENTRE 1 MATCH : TOT 1 = T 3
1970  = 1 K
1980 L = INT (K/N)
1990 J = K - L (N)
2000  IF L < 1 OR J < 1 OR J > N THEN 2020
2030  RETURN
2039  REM ENTREE DU CLOTE AU BAND
2040  REM ON ENTRE 1 MATCH : TOT 1 = T 3
2050  = 1 K
2060 L = INT (K/N)
2070 J = K - L (N)
2080  IF L < 1 OR J < 1 OR J > N THEN 2100
2130  RETURN
2139  REM ENTREE DU CLOTE AU BAND
2140  REM ON ENTRE 1 MATCH : TOT 1 = T 3
2150  = 1 K
2160 L = INT (K/N)
2170 J = K - L (N)
2180  IF L < 1 OR J < 1 OR J > N THEN 2200
2230  RETURN
2239  REM ENTREE DU CLOTE AU BAND
2240  REM ON ENTRE 1 MATCH : TOT 1 = T 3
2250  = 1 K
2260 L = INT (K/N)
2270 J = K - L (N)
2280  IF L < 1 OR J < 1 OR J > N THEN 2300
2330  RETURN
2339  REM ENTREE DU CLOTE AU BAND
2340  REM ON ENTRE 1 MATCH : TOT 1 = T 3
2350  = 1 K
2360 L = INT (K/N)
2370 J = K - L (N)
2380  IF L < 1 OR J < 1 OR J > N THEN 2400
2430  RETURN
2439  REM ENTREE DU CLOTE AU BAND
2440  REM ON ENTRE 1 MATCH : TOT 1 = T 3
2450  = 1 K
2460 L = INT (K/N)
2470 J = K - L (N)
2480  IF L < 1 OR J < 1 OR J > N THEN 2500
2530  RETURN
2539  REM ENTREE DU CLOTE AU BAND
2540  REM ON ENTRE 1 MATCH : TOT 1 = T 3
2550  = 1 K
2560 L = INT (K/N)
2570 J = K - L (N)
2580  IF L < 1 OR J < 1 OR J > N THEN 2600
2630  RETURN
2639  REM ENTREE DU CLOTE AU BAND
2640  REM ON ENTRE 1 MATCH : TOT 1 = T 3
2650  = 1 K
2660 L = INT (K/N)
2670 J = K - L (N)
2680  IF L < 1 OR J < 1 OR J > N THEN 2700
2730  RETURN
2739  REM ENTREE DU CLOTE AU BAND
2740  REM ON ENTRE 1 MATCH : TOT 1 = T 3
2750  = 1 K
2760 L = INT (K/N)
2770 J = K - L (N)
2780  IF L < 1 OR J < 1 OR J > N THEN 2800
2830  RETURN
2839  REM ENTREE DU CLOTE AU BAND
2840  REM ON ENTRE 1 MATCH : TOT 1 = T 3
2850  = 1 K
2860 L = INT (K/N)
2870 J = K - L (N)
2880  IF L < 1 OR J < 1 OR J > N THEN 2900
2930  RETURN
2939  REM ENTREE DU CLOTE AU BAND
2940  REM ON ENTRE 1 MATCH : TOT 1 = T 3
2950  = 1 K
2960 L = INT (K/N)
2970 J = K - L (N)
2980  IF L < 1 OR J < 1 OR J > N THEN 3000
3030  RETURN
3039  REM ENTREE DU CLOTE AU BAND
3040  REM ON ENTRE 1 MATCH : TOT 1 = T 3
3050  = 1 K
3060 L = INT (K/N)
3070 J = K - L (N)
3080  IF L < 1 OR J < 1 OR J > N THEN 3100
3130  RETURN
3139  REM ENTREE DU CLOTE AU BAND
3140  REM ON ENTRE 1 MATCH : TOT 1 = T 3
3150  = 1 K
3160 L = INT (K/N)
3170 J = K - L (N)
3180  IF L < 1 OR J < 1 OR J > N THEN 3200
3230  RETURN
3239  REM ENTREE DU CLOTE AU BAND
3240  REM ON ENTRE 1 MATCH : TOT 1 = T 3
3250  = 1 K
3260 L = INT (K/N)
3270 J = K - L (N)
3280  IF L < 1 OR J < 1 OR J > N THEN 3300
3330  RETURN
3339  REM ENTREE DU CLOTE AU BAND
3340  REM ON ENTRE 1 MATCH : TOT 1 = T 3
3350  = 1 K
3360 L = INT (K/N)
3370 J = K - L (N)
3380  IF L < 1 OR J < 1 OR J > N THEN 3400
3430  RETURN
3439  REM ENTREE DU CLOTE AU BAND
3440  REM ON ENTRE 1 MATCH : TOT 1 = T 3
3450  = 1 K
3460 L = INT (K/N)
3470 J = K - L (N)
3480  IF L < 1 OR J < 1 OR J > N THEN 3500
3530  RETURN
3539  REM ENTREE DU CLOTE AU BAND
3540  REM ON ENTRE 1 MATCH : TOT 1 = T 3
3550  = 1 K
3560 L = INT (K/N)
3570 J = K - L (N)
3580  IF L < 1 OR J < 1 OR J > N THEN 3600
3630  RETURN
3639  REM ENTREE DU CLOTE AU BAND
3640  REM ON ENTRE 1 MATCH : TOT 1 = T 3
3650  = 1 K
3660 L = INT (K/N)
3670 J = K - L (N)
3680  IF L < 1 OR J < 1 OR J > N THEN 3700
3730  RETURN
3739  REM ENTREE DU CLOTE AU BAND
3740  REM ON ENTRE 1 MATCH : TOT 1 = T 3
3750  = 1 K
3760 L = INT (K/N)
3770 J = K - L (N)
3780  IF L < 1 OR J < 1 OR J > N THEN 3800
3830  RETURN
3839  REM ENTREE DU CLOTE AU BAND
3840  REM ON ENTRE 1 MATCH : TOT 1 = T 3
3850  = 1 K
3860 L = INT (K/N)
3870 J = K - L (N)
3880  IF L < 1 OR J < 1 OR J > N THEN 3900
3930  RETURN
3939  REM ENTREE DU CLOTE AU BAND
3940  REM ON ENTRE 1 MATCH : TOT 1 = T 3
3950  = 1 K
3960 L = INT (K/N)
3970 J = K - L (N)
3980  IF L < 1 OR J < 1 OR J > N THEN 4000
4030  RETURN
4039  REM ENTREE DU CLOTE AU BAND
4040  REM ON ENTRE 1 MATCH : TOT 1 = T 3
4050  = 1 K
4060 L = INT (K/N)
4070 J = K - L (N)
4080  IF L < 1 OR J < 1 OR J > N THEN 4100
4130  RETURN
4139  REM ENTREE DU CLOTE AU BAND
4140  REM ON ENTRE 1 MATCH : TOT 1 = T 3
4150  = 1 K
4160 L = INT (K/N)
4170 J = K - L (N)
4180  IF L < 1 OR J < 1 OR J > N THEN 4200
4230  RETURN
4239  REM ENTREE DU CLOTE AU BAND
4240  REM ON ENTRE 1 MATCH : TOT 1 = T 3
4250  = 1 K
4260 L = INT (K/N)
4270 J = K - L (N)
4280  IF L < 1 OR J < 1 OR J > N THEN 4300
4330  RETURN
4339  REM ENTREE DU CLOTE AU BAND
4340  REM ON ENTRE 1 MATCH : TOT 1 = T 3
4350  = 1 K
4360 L = INT (K/N)
4370 J = K - L (N)
4380  IF L < 1 OR J < 1 OR J > N THEN 4400
4430  RETURN
4439  REM ENTREE DU CLOTE AU BAND
4440  REM ON ENTRE 1 MATCH : TOT 1 = T 3
4450  = 1 K
4460 L = INT (K/N)
4470 J = K - L (N)
4480  IF L < 1 OR J < 1 OR J > N THEN 4500
4530  RETURN
4539  REM ENTREE DU CLOTE AU BAND
4540  REM ON ENTRE 1 MATCH : TOT 1 = T 3
4550  = 1 K
4560 L = INT (K/N)
4570 J = K - L (N)
4580  IF L < 1 OR J < 1 OR J > N THEN 4600
4630  RETURN
4639  REM ENTREE DU CLOTE AU BAND
4640  REM ON ENTRE 1 MATCH : TOT 1 = T 3
4650  = 1 K
4660 L = INT (K/N)
4670 J = K - L (N)
4680  IF L < 1 OR J < 1 OR J > N THEN 4700
4730  RETURN
4739  REM ENTREE DU CLOTE AU BAND
4740  REM ON ENTRE 1 MATCH : TOT 1 = T 3
4750  = 1 K
4760 L = INT (K/N)
4770 J = K - L (N)
4780  IF L < 1 OR J < 1 OR J > N THEN 4800
4830  RETURN
4839  REM ENTREE DU CLOTE AU BAND
4840  REM ON ENTRE 1 MATCH : TOT 1 = T 3
4850  = 1 K
4860 L = INT (K/N)
4870 J = K - L (N)
4880  IF L < 1 OR J < 1 OR J > N THEN 4900
4930  RETURN
4939  REM ENTREE DU CLOTE AU BAND
4940  REM ON ENTRE 1 MATCH : TOT 1 = T 3
4950  = 1 K
4960 L = INT (K/N)
4970 J = K - L (N)
4980  IF L < 1 OR J < 1 OR J > N THEN 5000
5030  RETURN
5039  REM ENTREE DU CLOTE AU BAND
5040  REM ON ENTRE 1 MATCH : TOT 1 = T 3
5050  = 1 K
5060 L = INT (K/N)
5070 J = K - L (N)
5080  IF L < 1 OR J < 1 OR J > N THEN 5100
5130  RETURN
5139  REM ENTREE DU CLOTE AU BAND
5140  REM ON ENTRE 1 MATCH : TOT 1 = T 3
5150  = 1 K
5160 L = INT (K/N)
5170 J = K - L (N)
5180  IF L < 1 OR J < 1 OR J > N THEN 5200
5230  RETURN
5239  REM ENTREE DU CLOTE AU BAND
5240  REM ON ENTRE 1 MATCH : TOT 1 = T 3
5250  = 1 K
5260 L = INT (K/N)
5270 J = K - L (N)
5280  IF L < 1 OR J < 1 OR J > N THEN 5300
5330  RETURN
5339  REM ENTREE DU CLOTE AU BAND
5340  REM ON ENTRE 1 MATCH : TOT 1 = T 3
5350  = 1 K
5360 L = INT (K/N)
5370 J = K - L (N)
5380  IF L < 1 OR J < 1 OR J > N THEN 5400
5430  RETURN
5439  REM ENTREE DU CLOTE AU BAND
5440  REM ON ENTRE 1 MATCH : TOT 1 = T 3
5450  = 1 K
5460 L = INT (K/N)
5470 J = K - L (N)
5480  IF L < 1 OR J < 1 OR J > N THEN 5500
5530  RETURN
5539  REM ENTREE DU CLOTE AU BAND
5540  REM ON ENTRE 1 MATCH : TOT 1 = T 3
5550  = 1 K
5560 L = INT (K/N)
5570 J = K - L (N)
5580  IF L < 1 OR J < 1 OR J > N THEN 5600
5630  RETURN
5639  REM ENTREE DU CLOTE AU BAND
5640  REM ON ENTRE 1 MATCH : TOT 1 = T 3
5650  = 1 K
5660 L = INT (K/N)
5670 J = K - L (N)
5680  IF L < 1 OR J < 1 OR J > N THEN 5700
5730  RETURN
5739  REM ENTREE DU CLOTE AU BAND
5740  REM ON ENTRE 1 MATCH : TOT 1 = T 3
5750  = 1 K
5760 L = INT (K/N)
5770 J = K - L (N)
5780  IF L < 1 OR J < 1 OR J > N THEN 5800
5830  RETURN
5839  REM ENTREE DU CLOTE AU BAND
5840  REM ON ENTRE 1 MATCH : TOT 1 = T 3
5850  = 1 K
5860 L = INT (K/N)
5870 J = K - L (N)
5880  IF L < 1 OR J < 1 OR J > N THEN 5900
5930  RETURN
5939  REM ENTREE DU CLOTE AU BAND
5940  REM ON ENTRE 1 MATCH : TOT 1 = T 3
5950  = 1 K
5960 L = INT (K/N)
5970 J = K - L (N)
5980  IF L < 1 OR J < 1 OR J > N THEN 6000
6030  RETURN
6039  REM ENTREE DU CLOTE AU BAND
6040  REM ON ENTRE 1 MATCH : TOT 1 = T 3
6050  = 1 K
6060 L = INT (K/N)
6070 J = K - L (N)
6080  IF L < 1 OR J < 1 OR J > N THEN 6100
6130  RETURN
6139  REM ENTREE DU CLOTE AU BAND
6140  REM ON ENTRE 1 MATCH : TOT 1 = T 3
6150  = 1 K
6160 L = INT (K/N)
6170 J = K - L (N)
6180  IF L < 1 OR J < 1 OR J > N THEN 6200
6230  RETURN
6239  REM ENTREE DU CLOTE AU BAND
6240  REM ON ENTRE 1 MATCH : TOT 1 = T 3
6250  = 1 K
6260 L = INT (K/N)
6270 J = K - L (N)
6280  IF L < 1 OR J < 1 OR J > N THEN 6300
6330  RETURN
6339  REM ENTREE DU CLOTE AU BAND
6340  REM ON ENTRE 1 MATCH : TOT 1 = T 3
6350  = 1 K
6360 L = INT (K/N)
6370 J = K - L (N)
6380  IF L < 1 OR J < 1 OR J > N THEN 6400
6430  RETURN
6439  REM ENTREE DU CLOTE AU BAND
6440  REM ON ENTRE 1 MATCH : TOT 1 = T 3
6450  = 1 K
6460 L = INT (K/N)
6470 J = K - L (N)
6480  IF L < 1 OR J < 1 OR J > N THEN 6500
6530  RETURN
6539  REM ENTREE DU CLOTE AU BAND
6540  REM ON ENTRE 1 MATCH : TOT 1 = T 3
6550  = 1 K
6560 L = INT (K/N)
6570 J = K - L (N)
6580  IF L < 1 OR J < 1 OR J > N THEN 6600
6630  RETURN
6639  REM ENTREE DU CLOTE AU BAND
6640  REM ON ENTRE 1 MATCH : TOT 1 = T 3
6650  = 1 K
6660 L = INT (K/N)
6670 J = K - L (N)
6680  IF L < 1 OR J < 1 OR J > N THEN 6700
6730  RETURN
6739  REM ENTREE DU CLOTE AU BAND
6740  REM ON ENTRE 1 MATCH : TOT 1 = T 3
6750  = 1 K
6760 L = INT (K/N)
6770 J = K - L (N)
6780  IF L < 1 OR J < 1 OR J > N THEN 6800
6830  RETURN
6839  REM ENTREE DU CLOTE AU BAND
6840  REM ON ENTRE 1 MATCH : TOT 1 = T 3
6850  = 1 K
6860 L = INT (K/N)
6870 J = K - L (N)
6880  IF L < 1 OR J < 1 OR J > N THEN 6900
6930  RETURN
6939  REM ENTREE DU CLOTE AU BAND
6940  REM ON ENTRE 1 MATCH : TOT 1 = T 3
6950  = 1 K
6960 L = INT (K/N)
6970 J = K - L (N)
6980  IF L < 1 OR J < 1 OR J > N THEN 7000
7030  RETURN
7039  REM ENTREE DU CLOTE AU BAND
7040  REM ON ENTRE 1 MATCH : TOT 1 = T 3
7050  = 1 K
7060 L = INT (K/N)
7070 J = K - L (N)
7080  IF L < 1 OR J < 1 OR J > N THEN 7100
7130  RETURN
7139  REM ENTREE DU CLOTE AU BAND
7140  REM ON ENTRE 1 MATCH : TOT 1 = T 3
7150  = 1 K
7160 L = INT (K/N)
7170 J = K - L (N)
7180  IF L < 1 OR J < 1 OR J > N THEN 7200
7230  RETURN
7239  REM ENTREE DU CLOTE AU BAND
7240  REM ON ENTRE 1 MATCH : TOT 1 = T 3
7250  = 1 K
7260 L = INT (K/N)
7270 J = K - L (N)
7280  IF L < 1 OR J < 1 OR J > N THEN 7300
7330  RETURN
7339  REM ENTREE DU CLOTE AU BAND
7340  REM ON ENTRE 1 MATCH : TOT 1 = T 3
7350  = 1 K
7360 L = INT (K/N)
7370 J = K - L (N)
7380  IF L < 1 OR J < 1 OR J > N THEN 7400
7430  RETURN
7439  REM ENTREE DU CLOTE AU BAND
7440  REM ON ENTRE 1 MATCH : TOT 1 = T 3
7450  = 1 K
7460 L = INT (K/N)
7470 J = K - L (N)
7480  IF L < 1 OR J < 1 OR J > N THEN 7500
7530  RETURN
7539  REM ENTREE DU CLOTE AU BAND
7540  REM ON ENTRE 1 MATCH : TOT 1 = T 3
7550  = 1 K
7560 L = INT (K/N)
7570 J = K - L (N)
7580  IF L < 1 OR J < 1 OR J > N THEN 7600
7630  RETURN
7639  REM ENTREE DU CLOTE AU BAND
7640  REM ON ENTRE 1 MATCH : TOT 1 = T 3
7650  = 1 K
7660 L = INT (K/N)
7670 J = K - L (N)
7680  IF L < 1 OR J < 1 OR J > N THEN 7700
7730  RETURN
7739  REM ENTREE DU CLOTE AU BAND
7740  REM ON ENTRE 1 MATCH : TOT 1 = T 3
7750  = 1 K
7760 L = INT (K/N)
7770 J = K - L (N)
7780  IF L < 1 OR J < 1 OR J > N THEN 7800
7830  RETURN
7839  REM ENTREE DU CLOTE AU BAND
7840  REM ON ENTRE 1 MATCH : TOT 1 = T 3
7850  = 1 K
7860 L = INT (K/N)
7870 J = K - L (N)
7880  IF L < 1 OR J < 1 OR J > N THEN 7900
7930  RETURN
7939  REM ENTREE DU CLOTE AU BAND
7940  REM ON ENTRE 1 MATCH : TOT 1 = T 3
7950  = 1 K
7960 L = INT (K/N)
7970 J = K - L (N)
7980  IF L < 1 OR J < 1 OR J > N THEN 8000
8030  RETURN
8039  REM ENTREE DU CLOTE AU BAND
8040  REM ON ENTRE 1 MATCH : TOT 1 = T 3
8050  = 1 K
8060 L = INT (K/N)
8070 J = K - L (N)
8080  IF L < 1 OR J < 1 OR J > N THEN 8100
8130  RETURN
8139  REM ENTREE DU CLOTE AU BAND
8140  REM ON ENTRE 1 MATCH : TOT 1 = T 3
8150  = 1 K
8160 L = INT (K/N)
8170 J = K - L (N)
8180  IF L < 1 OR J < 1 OR J > N THEN 8200
8230  RETURN
8239  REM ENTREE DU CLOTE AU BAND
8240  REM ON ENTRE 1 MATCH : TOT 1 = T 3
8250  = 1 K
8260 L = INT (K/N)
8270 J = K - L (N)
8280  IF L < 1 OR J < 1 OR J > N THEN 8300
8330  RETURN
8339  REM ENTREE DU CLOTE AU BAND
8340  REM ON ENTRE 1 MATCH : TOT 1 = T 3
8350  = 1 K
8360 L = INT (K/N)
8370 J = K - L (N)
8380  IF L < 1 OR J < 1 OR J > N THEN 8400
8430  RETURN
8439  REM ENTREE DU CLOTE AU BAND
8440  REM ON ENTRE 1 MATCH : TOT 1 = T 3
8450  = 1 K
8460 L = INT (K/N)
8470 J = K - L (N)
8480  IF L < 1 OR J < 1 OR J > N THEN 8500
8530  RETURN
8539  REM ENTREE DU CLOTE AU BAND
8540  REM ON ENTRE 1 MATCH : TOT 1 = T 3
8550  = 1 K
8560 L = INT (K/N)
8570 J = K - L (N)
8580  IF L < 1 OR J < 1 OR J > N THEN 8600
8630  RETURN
8639  REM ENTREE DU CLOTE AU BAND
8640  REM ON ENTRE 1 MATCH : TOT 1 = T 3
8650  = 1 K
8660 L = INT (K/N)
8670 J = K - L (N)
8680  IF L < 1 OR J < 1 OR J > N THEN 8700
8730  RETURN
8739  REM ENTREE DU CLOTE AU BAND
8740  REM ON ENTRE 1 MATCH : TOT 1 = T 3
8750  = 1 K
8760 L = INT (K/N)
8770 J = K - L (N)
8780  IF L < 1 OR J < 1 OR J > N THEN 8800
8830  RETURN
8839  REM ENTREE DU CLOTE AU BAND
8840  REM ON ENTRE 1 MATCH : TOT 1 = T 3
8850  = 1 K
8860 L = INT (K/N)
8870 J = K - L (N)
8880  IF L < 1 OR J < 1 OR J > N THEN 8900
8930  RETURN
8939  REM ENTREE DU CLOTE AU BAND
8940  REM ON ENTRE 1 MATCH : TOT 1 = T 3
8950  = 1 K
8960 L = INT (K/N)
8970 J = K - L (N)
8980  IF L < 1 OR J < 1 OR J > N THEN 9000
9030  RETURN
9039  REM ENTREE DU CLOTE AU BAND
9040  REM ON ENTRE 1 MATCH : TOT 1 = T 3
9050  = 1 K
9060 L = INT (K/N)
9070 J = K - L (N)
9080  IF L < 1 OR J < 1 OR J > N THEN 9100
9130  RETURN
9139  REM ENTREE DU CLOTE AU BAND
9140  REM ON ENTRE 1 MATCH : TOT 1 = T 3
9150  = 1 K
9160 L = INT (K/N)
9170 J = K - L (N)
9180  IF L < 1 OR J < 1 OR J > N THEN 9200
9230  RETURN
9239  REM ENTREE DU CLOTE AU BAND
9240  REM ON ENTRE 1 MATCH : TOT 1 = T 3
9250  = 1 K
9260 L = INT (K/N)
9270 J = K - L (N)
9280  IF L < 1 OR J < 1 OR J > N THEN 9300
9330  RETURN
9339  REM ENTREE DU CLOTE AU BAND
9340  REM ON ENTRE 1 MATCH : TOT 1 = T 3
9350  = 1 K
9360 L = INT (K/N)
9370 J = K - L (N)
9380  IF L < 1 OR J < 1 OR J > N THEN 9400
9430  RETURN
9439  REM ENTREE DU CLOTE AU BAND
9440  REM ON ENTRE 1 MATCH : TOT 1 = T 3
9450  = 1 K
9460 L = INT (K/N)
9470 J = K - L (N)
9480  IF L < 1 OR J < 1 OR J > N THEN 9500
9530  RETURN
9539  REM ENTREE DU CLOTE AU BAND
9540  REM ON ENTRE 1 MATCH : TOT 1 = T 3
9550  = 1 K
9560 L = INT (K/N)
9570 J = K - L (N)
9580  IF L < 1 OR J < 1 OR J > N THEN 9600
9630  RETURN
9639  REM ENTREE DU CLOTE AU BAND
9640  REM ON ENTRE 1 MATCH : TOT 1 = T 3
9650  = 1 K
9660 L = INT (K/N)
9670 J = K - L (N)
9680  IF L < 1 OR J < 1 OR J > N THEN 9700
9730  RETURN
9739  REM ENTREE DU CLOTE AU BAND
9740  REM ON ENTRE 1 MATCH : TOT 1 = T 3
9750  = 1 K
9760 L = INT (K/N)
9770 J = K - L (N)
9780  IF L < 1 OR J < 1 OR J > N THEN 9800
9830  RETURN
9839  REM ENTREE DU CLOTE AU BAND
9840  REM ON ENTRE 1 MATCH : TOT 1 = T 3
9850  = 1 K
9860 L = INT (K/N)
9870 J = K - L (N)
9880  IF L < 1 OR J < 1 OR J > N THEN 9900
9930  RETURN
9939  REM ENTREE DU CLOTE AU BAND
9940  REM ON ENTRE 1 MATCH : TOT 1 = T 3
9950  = 1 K
9960 L = INT (K/N)
9970 J = K - L (N)
9980  IF L < 1 OR J < 1 OR J > N THEN 10000
10030  RETURN
10039  REM ENTREE DU CLOTE AU BAND
10040  REM ON ENTRE 1 MATCH : TOT 1 = T 3
10050  = 1 K
10060 L = INT (K/N)
10070 J = K - L (N)
10080  IF L < 1 OR J < 1 OR J > N THEN 10100
10130  RETURN
10139  REM ENTREE DU CLOTE AU BAND
10140  REM ON ENTRE 1 MATCH : TOT 1 = T 3
10150  = 1 K
10160 L = INT (K/N)
10170 J = K - L (N)
10180  IF L < 1 OR J < 1 OR J > N THEN 10200
10230  RETURN
10239  REM ENTREE DU CLOTE AU BAND
10240  REM ON ENTRE 1 MATCH : TOT 1 = T 3
10250  = 1 K
10260 L = INT (K/N)
10270 J = K - L (N)
10280  IF L < 1 OR J < 1 OR J > N THEN 10300
10330  RETURN
10339  REM ENTREE DU CLOTE AU BAND
10340  REM ON ENTRE 1 MATCH : TOT 1 = T 3
10350  = 1 K
10360 L = INT (K/N)
10370 J = K - L (N)
10380  IF L < 1 OR J < 1 OR J > N THEN 10400
10430  RETURN
10439  REM ENTREE DU CLOTE AU BAND
10440  REM ON ENTRE 1 MATCH : TOT 1 = T 3
10450  = 1 K
10460 L = INT (K/N)
10470 J = K - L (N)
10480  IF L < 1 OR J < 1 OR J > N THEN 10500
10530  RETURN
10539  REM ENTREE DU CLOTE AU BAND
10540  REM ON ENTRE 1 MATCH : TOT 1 = T 3
10550  = 1 K
10560 L = INT (K/N)
10570 J = K - L (N)
10580  IF L < 1 OR J < 1 OR J > N THEN 10600
10630  RETURN
10639  REM ENTREE DU CLOTE AU BAND
10640  REM ON ENTRE 1 MATCH : TOT 1 = T 3
10650  = 1 K
10660 L = INT (K/N)
10670 J = K - L (N)
10680  IF L < 1 OR J < 1 OR J > N THEN 
```

```
5140  NEXT J
5150  NEXT L
5160  GOTO 7900
5170  INPUT "QUEL MATCH ?",B5,B15
5180  L = 0 : J = 0
5190  FOR K = 1 TO N
5200  IF B4 = CQ1(K) THEN L = K
5210  IF B16 = B1$(K) THEN J = K
5220  NEXT K
5230  IF L = 0 OR L = J = 0 THEN PRINT "ERREUR !" : GOSUB 1500 : GOTO 5170
5240  IF MA(L,J) < 0 THEN N = - MA(L,J) : GOSUB 2900 : GOSUB 3200 : GOTO 5200
5250  Z = PP DEC((MA(L,J) - INT (MA(L,J))) * 100) : GOSUB 3300
5260  PRINT "ATTENTION ! A ACHA UN RESULTAT POUR ", PRINT "DE MATCH " : GOSUB 3300 : PRINT
5270  PRINT "VOULEZ-VOUS LE CHANGER ?" : GOSUB 1000 : IF A$ = "O" THEN GOSUB 3700
5280  GOTO 7900
5290  REM AFFICHAGE DES RESULTATS
6000  HOME
6010  ONSUB 3200
6020  GOSUB 4300
6030  PRINT TAB( 13) "RESULTATS :" : PRINT
6040  FOR L = 1 TO N
6050  FOR J = 1 TO N
6060  IF L = J THEN NEXT
6070  GOSUB 3100
6080  NEXT J
6090  NEXT L
6100  GOTO 7900
6110  REM RESULTAT D UN MATCH
7000  HOME
7010  INPUT "QUEL MATCH ?",B5,B15
7020  L = 0 : J = 0
7030  FOR K = 1 TO N
7040  IF B4 = B5$(K) THEN L = K
7050  IF B15 = B1$(K) THEN J = K
7060  NEXT K
7070  IF L = 0 OR L = J THEN PRINT "ERREUR !" : GOSUB 1500 : GOTO 7010
7080  IF MA(L,J) < 0 THEN N = - MA(L,J)
7100  IF MA(L,J) > 0 THEN N = INT (MA(L,J)) * 100
7110  GOSUB 2900
7120  PRINT TAB( 10) "RESULTATS" : PRINT
7130  GOSUB 3100
7140  REM POINT D'ENTREE SPECIAL DU 500
7150  PRINT : PRINT "VOULEZ-VOUS LE CLASSEMENT A L'ESSUE DE" : PRINT "CETTE JOURNEE ?"
7160  GOSUB 1000
7170  IF A$ = "N" THEN RETURN
7180  IF N < CL1(0,5) THEN N=20
7190  REM AFFICHAGE DU CLASSEMENT
8000  HOME
8010  X = CL1(0,5) : IF X = 0 THEN X = 1
8020  IF CL1(0,5) < > N THEN GOSUB 900
8030  HOME
8040  PRINT "  CLASSEMENT      PTS  J  G  N  P  +  -"
8050  PRINT
8060  FOR L = 1 TO N
8070  IF CL1(L,1) = CL1(L - 1,1) THEN A$ = "  " : GOTO 8090
8080  A$ = STR$ (L) + " " : IF L < 10 THEN A$ = " " + A$
8090  PRINT A$;
8100  PRINT LEFT$ (B1$(CL1(L,1)) + LL$,14) ;
8110  IF CL1(L,1) < 10 THEN PRINT " "
8120  PRINT LL$(L,1) " ";
8130  W = LL$(L,2) + CL1(L,2) + CL1(L,4) : REM JOURS
8140  IF W < 10 THEN PRINT " "
8150  PRINT W " ";
8160  FOR J = 2 TO 6
8170  IF CL1(L,J) < 10 THEN PRINT " ";
8180  PRINT CL1(L,J) " "
8190  NEXT J
8200  PRINT
8210  NEXT L
8220  PRINT
8230  PRINT "ON PEUT REVENIR AU MENU ?"
8240  GOSUB 1000
8250  RETURN
9000  REM DATAS
9010  DATA 20  REM NOMBRE D'EQUIPES
9020  DATA ANGERS,AUXERRE
9030  DATA BASTIA,BORDEAUX
9040  DATA LAVAL,LENS,LILLE,LYON
9050  DATA METZ,MONACO
9060  DATA NANCY,NANTES,NICE,NIMES
9070  DATA PARIS SPG
9080  DATA ST-ETIENNE,STADE STRASBOURG
9090  DATA TOULOUSE
9100  DATA VALENCIENNES
9999  REM FIN
10000  HOME
10010  IF MA(0,0) = 1 THEN  GOSUB 1500
10020  END
10999  REM CALENDRIER D UNE ROUTE
11000  HOME
11010  INPUT "QUELLE EQUIPE ?" B6
11020  K = 0
11030  FOR L = 1 TO N
11040  IF B1$(L) = B6 THEN K = L : L = N
11050  NEXT L
11060  IF K = 0 THEN  PRINT "ERREUR" : GOSUB 1500 : GOTO 11010
11070  HOME
11080  PRINT  CALENDRIER DE L'EQUIPE DE " B1$(K)
11090  PRINT "MATCHS ALLER" : PRINT
11100  L = 0
11110  FOR A = 1 TO N - 1
11120  FOR J = 1 TO N
11130  GOSUB 3100 : S = L,K = J,J = L
11140  GOSUB 3100 : S = L,K = J,J = S
11150  NEXT J
11160  NEXT B
11170  PRINT : PRINT "VOULEZ VOUS VOIR LES MATCHS RETOUR ?" : GOSUB 1000  IF A$ = "O" THEN 11200
11180  HOME : PRINT  CALENDRIER DE L'EQUIPE DE " B1$(K)
11190  PRINT "MATCHS RETOUR" : PRINT
11200  L = 0
11210  FOR A = N TO N * 2 - 2
11220  FOR L = 1 TO N
11230  GOSUB 3100 : S = L,K = J,J = S
11240  GOSUB 3100 : S = L,K = J,J = S
11250  NEXT L
11260  NEXT B
11270  PRINT : PRINT "ON PEUT REVENIR AU MENU ?" : GOSUB 1000
11280  RETURN
11290  END
11300  REM ROUTINE SPECIALE DE MAINTENANCE
11310  REM ACCES PAR RUN 12000
12000  HOME : GOSUB 4300
12010  INPUT "MATCH ?",B4,B15
12020  L = 0 : J = 0
12030  FOR A = 1 TO N
12040  IF B4 = B1$(A) THEN L = A
12050  IF B15 = B1$(A) THEN J = A
12060  NEXT A
12070  IF L = J = 0 THEN  PRINT "ERREUR" : GOSUB 1500  GOTO 12010
12080  A = MA(L,J)
12090  PRINT "MATCH " B1$(L),"  " B1$(J) : GOSUB 2910
12100  PRINT "A CHANGE (O/N) ?" : GOSUB 1000  IF A$ = "N" THEN 12130
12110  INPUT "NOUVEL RESULTAT ?",A
12120  MA(L,J) = - INT(A) * 1
12130  PRINT "ENTRE (O/Q ?" : GOSUB 1000 : IF A$ = "O" THEN 12010
12140  GOTO 10
12150  REM ATTENTION ! IL AURA FALLUS RE UN DEBANCHE DU MATCH DEJA JOUE !
```

Ce sous-programme sauve le tableau des matches sur disque si on l'a modifié. Il se termine par un END et non par RETURN. Sur l'Apple, après avoir sélectionné l'option fin, il est possible de revenir au programme en tapant RETURN avec toutes données intactes. Cela permet une sauvegarde anticipée. Attention, le SED refusera d'effectuer une nouvelle sauvegarde si un « INPUT » n'a pas été effectué entre temps. Cela n'est pas gênant puisque l'option 1) utilise input.

27) calendrier d'une équipe

numéro : 11000
paramètres : —
résultat : —
appelle : 3100 (affiche une ligne de résultats)
1500 (liste des équipes)
appelle par : 0 (programme principal)
variables : L, J : boucles
R : numéro de l'équipe, et boucle
R$: équipe et réponse
K : échange (intermédiaire)

28) correction du calendrier

Ce sous-programme n'apparaît pas au menu. Il s'appelle par RUN 12000. Il s'emploie après 11000 quand on désire corriger des erreurs. Il s'agit en fait d'un programme dans le programme, un utilitaire de maintenance. Le programme et ses données tiennent dans 16 K de mémoire vive.

Le programme s'explique lui-même. Une seule remarque dans l'option 1) (entrée de résultats), l'entrée d'un nombre négatif pour l'un ou l'autre des scores (re)mettre le match correspondant « à jouer ». Cela permet la récupération des erreurs. Les noms des équipes et les scores doivent être séparés par une virgule ou par un retour chariot.

L'extension la plus immédiate est de rajouter au nom des fichiers l'année à laquelle ils s'appliquent. On peut ainsi constituer une véritable anthologie du championnat. Mais cela impose de lire le nom des équipes dans un fichier disque, puisqu'elles changent toutes les saisons. On peut voir plus loin en établissant des statistiques, ce qui mène à l'étude des gains, fonction des équipes et des conditions de jeu.

Enfin, l'ultime extension du programme consiste à lui faire établir des prévisions sur les résultats. On connaissait déjà les prévisions des résultats de tiercé par ordinateur, alors pourquoi pas le football...

Jean-Denis Muys

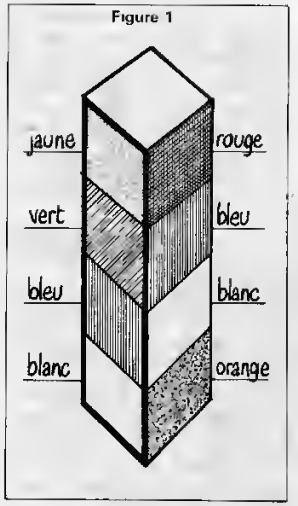
comment faire pour empiler des cubes

un problème résolu pas à pas

Il n'est pas possible de représenter tous les jeux par un programme d'ordinateur. Il est cependant amusant d'essayer en se promenant dans les rayons des grands magasins d'imaginer des algorithmes solutions des jeux exposés. Voici un programme qui résout un problème de cubes à empiler. Hélas, ce n'est qu'un algorithme et non un modèle du chemin suivi par un enfant qui joue.

Voici un jeu d'enfant ou presque. Il s'agit d'empiler quatre cubes, chaque cube possède six faces — en êtes-vous bien sûr ? — colorées, le jeu consiste à empiler les cubes de telle façon que si l'on considère les quatre *façades* verticales formées par l'empilement (et non les faces au-dessus et en dessous), sur chaque façade, n'apparaîtront que des couleurs différentes *(figure 1)*.

Figure 1



Les deux façades visibles de l'empilement présentent chacune quatre couleurs différentes

Bien sûr, si les quatre cubes sont entièrement rouges, il n'y aura aucune façon d'empiler les cubes qui réponde au problème, et inversement si les 24 faces des quatre cubes sont de 24 couleurs différentes, alors n'importe quel empilement sera une solution du problème. D'autre part, une même couleur peut apparaître plusieurs fois dans l'empilement, à condition, évidemment, que ce soit sur des façades différentes. Généralement, les couleurs des cubes que l'on trouve dans le commerce sont disposées de telle façon qu'une seule configuration est une solution (tout au moins à une rotation ou une symétrie près). Ou, recherches longues, très longues...

Ce problème n'en est un que pour les adultes, les enfants le considèrent plus facilement comme un jeu. On a remarqué, en effet, que les enfants trouvent généralement la solution plus rapidement que les adultes, en quelques minutes contre plusieurs heures.

Avant de faire l'analyse et le programme présentés ici, j'ai désespérément essayé — sans méthode — de jouer aux cubes. Heureusement, mon ordinateur individuel est venu à mon secours.

La résolution d'un tel problème est très typique d'une certaine forme de programmation *l'exploration d'arborescences*. Il se rapproche en cela de beaucoup d'autres jeux : le solitaire, le problème du placement sans prises réciproques de n dames sur un échiquier de côté n, le problème du voyageur de commerce : passer par « x » villes une seule fois en faisant le moins de kilomètres possible.

Contrairement aux échecs, il offre l'avantage de présenter une arborescence suffisamment réduite pour pouvoir être parcourue dans des temps humainement concevables. En effet, examinons comment nous pouvons empiler des cubes. Un cube possède six faces, lorsqu'il est posé sur une face donnée, on peut par quarts de tour successifs lui donner quatre positions différentes par rapport à un observateur fixe ou par rapport à un autre cube, et donc pour chacune des six faces

Première parution : L'OI n° 19

Table les 24 positions possibles d'un cube

1,2,3,4		POSITION DE DEPART
	2,3,4,1	ET POSITIONS DECALEES
	3,4,1,2	D'UN QUART DE TOUR
	4,1,2,3	
1,4,3,2		ON PERMUTE 2 ET 4
	4,3,2,1	
	3,2,1,4	
	2,1,4,3	
1,6,3,5		4 PASSE EN DESSOUS
	6,3,5,1	
	3,5,1,6	
	5,1,6,3	
1,5,3,6		ON PERMUTE 5 ET 6
	5,3,6,1	
	3,6,1,5	
	6,1,5,3	
2,5,4,6		METTONS 1 AU-DESSUS
	5,4,6,2	
	4,6,2,5	
	6,2,5,4	
2,6,4,5		METTONS 1 AU-DESSOUS
	6,4,5,2	
	4,5,2,6	
	5,2,6,4	

Les faces du dessus et du dessous sont déterminées par la donnée des quatre autres faces

sur lesquelles on peut poser le cube, il y a quatre possibilités de placement. Il existe donc 24 façons différentes de poser un cube sur un autre. Toutefois, pour ce qui est du premier cube, celui qui est posé sur le sol, nous ne considérons que 6 positions différentes : les quarts de tours qu'on pourrait faire faire à ce cube (et donc aux trois autres cubes empilés sur celui-ci) n'amèneront pas de solution nouvelle.

Pour chacune des 6 positions possibles du premier cube, il y a 24 façons d'empiler le deuxième cube, et pour chacune de ces 6 x 24 façons d'empiler deux cubes, il y a 24 façons d'empiler le troisième cube. De même pour le quatrième cube, ce qui amène à :

6 x 24 x 24 x 24
= 82944 façons

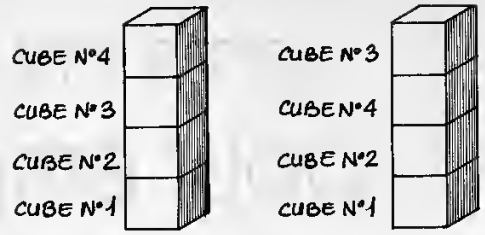
d'empiler quatre cubes. Ce n'est pas rien !

Il est donc à peu près certain qu'un enfant ne procède pas algorithmiquement pour trouver la solution de ce jeu. (Si quelqu'un a des idées à ce sujet, qu'il nous les communique !)

Quant à nous — et avec un petit ordinateur —, voyons comment explorer ce monde cubique et en extraire la ou les solutions.

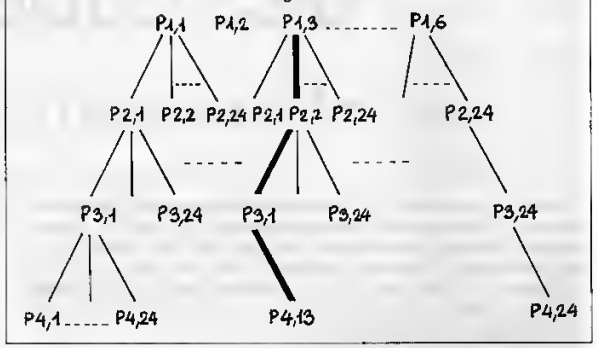
Appellons i-èmement correct un empilage de « i » cubes résolvant le problème. Par exemple un empilage de deux cubes tel qu'aucune façade verticale ne comporte de répétition de couleur sera dit « deuxièmement-correct », et la solution est un empilage quatrièmement-correct.

Figure 2



Deux solutions identiques

Figure 3



L'arbre représentant les 82 944 façons d'empiler les cubes

Il est par ailleurs évident que la position d'un cube donné dans la pile n'a aucune importance. Si un empilage est quatrièmement-correct, on peut, à condition de ne pas modifier leurs orientations réciproques, remplacer le cube le plus bas par le cube le plus haut, ou toute autre permutation du même type *(figure 2)*. C'est-à-dire que si 4-3-2-1 est solution, 3-4-2-1 l'est aussi, de même 1-2-3-4, etc.

L'idée de base est simple : empilons deux cubes de façon deuxièmement-correcte (il peut y avoir plusieurs empilages deuxièmement-corrects) et essayons d'ajouter un troisième cube pour obtenir un empilage troisièmement-correct. Si nous y arrivons, essayons d'ajouter le quatrième cube. Si aucune position du troisième cube ne répond à notre attente, il nous faut changer la position du deuxième cube et prendre un autre des empilages deuxièmement-corrects pour essayer à nouveau d'ajouter le troisième cube.

On peut représenter toutes les façons d'empiler quatre cubes par un arbre de positions. Rechercher *toutes* les solutions consistera alors à parcourir successivement *toutes* les branches de cet arbre (sauf en cas de rencontre d'une position ièmement incorrecte).

Soit P_{ij} la j-ème position possible du cube « i ». Dans notre problème, $1 \leq j \leq 24$ et $1 \leq i \leq 4$. On a donc le schéma de la *figure 3* qui représente tous les empilages possibles de 4 cubes. Sur cette figure, nous avons représenté en trait gras l'empilage correspondant à la position 3 du cube 1, 2 du cube 2, 1 du cube 3, et 13 du cube 4.

Les cubes dans la mémoire

Comment représenter tous nos cubes dans un ordinateur ? Appelons C1, C2, C3, C4 nos quatre cubes.

1. Numérotons les faces des cubes, et à chaque face associons une couleur. On aura donc pour le cube Ci une table T$(i) à 6 éléments contenant les couleurs des faces *(figure 4)*.

Exemple d'exécution

```
RENTREZ LES COULEURS DES FACES DU CUBE 1
FACE1, FACE2, ......,FACE6
? J,R,V,B,R,R

RENTREZ LES COULEURS DES FACES DU CUBE 2
? R,V,B,J,R,R

RENTREZ LES COULEURS DES FACES DU CUBE 3
? V,B,J,R,R,R

RENTREZ LES COULEURS DES FACES DU CUBE 4
? B,J,R,V,R,R
MERCI ....
```

```
SOLUTION NUMERO 1
                 CUBE 1  CUBE 2  CUBE 3  CUBE 4
FACADE 1 :        J       R       V       B
FACADE 2 :        R       V       B       J
FACADE 3 :        V       B       J       R
FACADE 4 :        B       J       R       V
SOLUTION NUMERO 2
                 CUBE 1  CUBE 2  CUBE 3  CUBE 4
FACADE 1 :        J       R       B       V
FACADE 2 :        R       V       J       B
FACADE 3 :        V       B       R       J
FACADE 4 :        B       J       V       R
SOLUTION NUMERO 3
                 CUBE 1  CUBE 2  CUBE 3  CUBE 4
FACADE 1 :        J       V       B       R
FACADE 2 :        R       B       J       V
FACADE 3 :        V       J       R       B
FACADE 4 :        B       R       V       J
SOLUTION NUMERO 4
                 CUBE 1  CUBE 2  CUBE 3  CUBE 4
FACADE 1 :        J       V       R       B
FACADE 2 :        R       B       V       J
FACADE 3 :        V       J       B       R
FACADE 4 :        B       R       J       V
```

Figure 4

T$(i)	1	2	3	4	5	6
	J	R	V	B	J	B
	face 1					face 6

La table des couleurs des 6 faces du cube 1, avec l'indication de certaines couleurs

Dans le programme, nous regrouperons en fait les 4 tables T$(i) dans un seul tableau T$(i, j) avec i compris entre 1 et 4, et j entre 1 et 8.

2. Il faut maintenant représenter une position d'un cube. Commençons par faire une croix sur la face qui se trouve le plus proche de nous, et un rond sur celle du dessus (seul moyen de nous y retrouver).
Le cube sera alors représenté comme sur la *figure 5*.

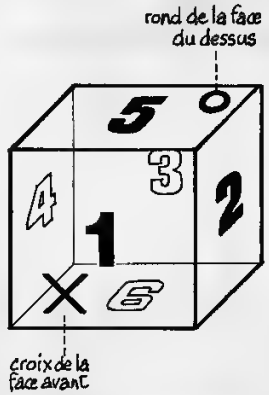
Remarques :

- Un quart de tour, c'est une permutation circulaire sur les contenus des quatre premières cases, par exemple, 1234 donne, de quart de tour en quart de tour, 2341, 3412 et 4123.
- Un retournement sens dessus-dessous, c'est une inversion des contenus des cases 5 et 6 (et, par conséquent, 2 et 4). Ainsi, 123456 devient 143265.

Nous pouvons donc construire une table appelée PO dans laquelle seront représentées les 24 positions possibles d'un cube numérotées de 1 à 24. Comme nous ne nous préoccupons pas des faces dessus-dessous, ne les représentons pas dans cette table.

Nous allons nous servir de PO pour représenter les positions de tous les cubes : la position d'un cube Ci à un instant donné est la valeur contenue dans la variable P(i). Cette valeur est un indice compris entre 1 et 24 de la table PO.

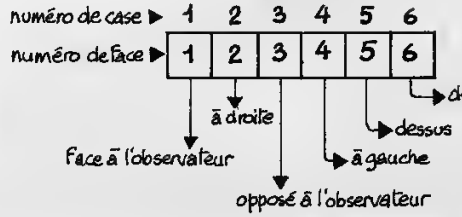
Figure 5



Par exemple, supposons le cube 1 dans la position numéro 4. Nous voulons connaître la couleur de la face opposée à l'observateur, sachant que les couleurs ont été attribuées dans T$(1) comme dans la *figure 4*. Allons à la ligne 4 de PO. Nous avons décidé plus haut que la face opposée à l'observateur serait dans la troisième case. Nous voyons donc que ce sera la face 8 du cube 1. Consultons T$(1), cette face est 8 (comme blanche).
Comment essayer toutes les positions possibles pour un cube « i » : on commence avec P(i) = 1 et on essaye cette position. Chaque fois qu'on veut une nouvelle position, on fait P(i) = P(i) + 1 (ou bien P(i) = P(i) + 4 pour le cube 1) et si on arrive à P(i) = 24, c'est que l'on a essayé toutes les positions.

Dira que deux cubes Ci et Cj dans des positions P(i) et P(j) sont dauxièmement-corrects, c'est dire que pour k valant 1, 2, 3 et 4 :
T$(i, (PO [P(i), k])) =
T$(j, (PO [P(j), k]))
d'où une procédure de vérification de « correction-dauxième » (empi-

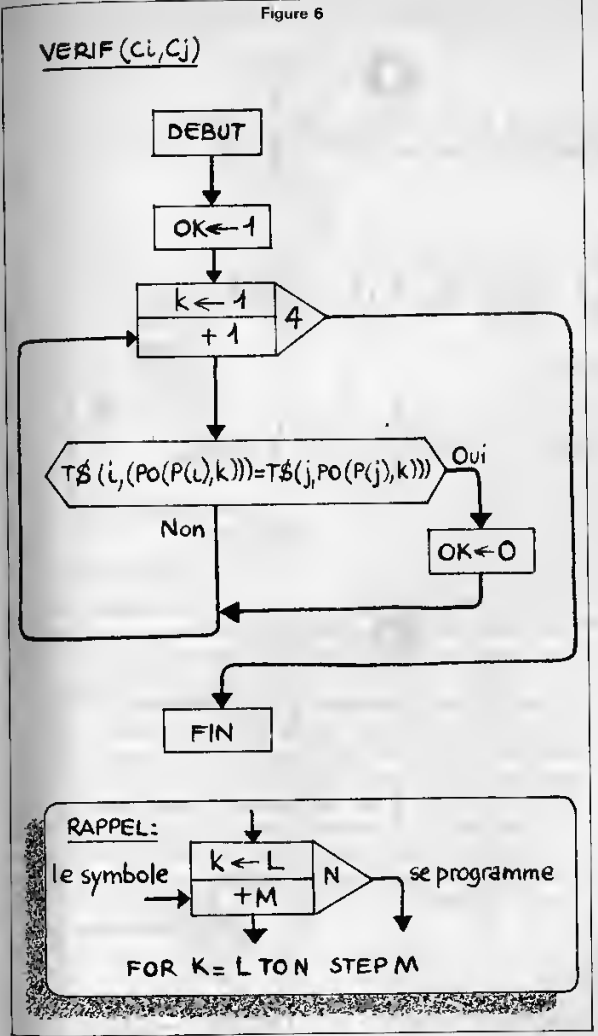
La numérotation des cases une fois le cube repéré, par la croix et le rond (ci-dessus) et le repérage en mémoire de la position (ci-dessous)



Liste du programme

```
10 REM          ***************************
20 REM          *     LES CUBES ERFLES    *
30 REM          ***************************
40 REM     COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
50 REM
60 REM ****************TABLE DES VARIABLES UTILISEES*******************
70 REM
80 REM
90 REM ==========================================================
100 REM     T$(6,6)     TABLEAU DES COULEURS DES 6 FACES DE CHACUN DES 4 CUBES
110 REM     P(4)        TABLEAU DES POSITIONS COURANTES DES 4 CUBES (UTILISE
120 REM                 DANS LES SOUS-PROG "VERIF")
130 REM     PO(24,4)    TABLEAU DES 24 POSITIONS POSSIBLES D'UN CUBE, AVEC
140 REM                 POUR CHAQUE POSITION LES 4 FACES VISIBLES
150 REM     W           COMPTEUR DU NOMBRE D'ESSAIS QUATRIEME-CORRECTS
160 REM     L,Y         DANS L'INITIALISATION, COMPTEURS POUR REMPLIR PO(L,4)
170 REM     Q1          INDICE DES POSITIONS SUCCESSIVES DE CUBE 1
180 REM     Q2          INDICE DES POSITIONS SUCCESSIVES DE CUBE 2
190 REM     Q3          INDICE DES POSITIONS SUCCESSIVES DE CUBE 3
200 REM     Q4          INDICE DES POSITIONS SUCCESSIVES DE CUBE4
210 REM     J,J         NUMEROS DES POSITIONS SUR LESQUELLES VA ETRE APPLIQUE
220 REM                 LE PROGRAMME VERIF
230 REM     OK          DRAPEAU RAMENANT LE RESULTAT DE L'APPEL DE
240 REM                 "VERIF (C2,C3)"-PRIND LES VALEURS 0 OU 1
250 REM     K           COMPTEUR DES FACES DANS PO
260 REM
270 REM ****************** INITIALISATIONS ***********************
280 DIM T$(6,6),P(4),PO(24,4)
290 PRINT "RENTREZ LES COULEURS DES FACES DU CUBE 1"
300 PRINT "RENTREZ LES COULEURS DES FACES DU CUBE 2"
310 INPUT T$(1,1),T$(1,2),T$(1,3),T$(1,4),T$(1,5),T$(1,6)
320 PRINT "RENTREZ LES COULEURS DES FACES DU CUBE 3"
330 INPUT T$(2,1),T$(2,2),T$(2,3),T$(2,4),T$(2,5),T$(2,6)
340 PRINT "RENTREZ LES COULEURS DES FACES DU CUBE 4"
350 INPUT T$(3,1),T$(3,2),T$(3,3),T$(3,4),T$(3,5),T$(3,6)
360 PRINT "RENTREZ LES COULEURS DES FACES DU CUBE 4"
370 INPUT T$(4,1),T$(4,2),T$(4,3),T$(4,4),T$(4,5),T$(4,6)
380 PRINT "PERCI ....
400 REM ---- LES 24 POSITIONS D'UN CUBE          REM POSITION DE DEPART
410 DATA        1,2,3,4
420 DATA        2,3,4,1 : REM ET POSITIONS DECALEES D'UN
430 DATA        3,4,1,2 :REM QUART DE TOUR
440 DATA        4,1,2,3
450 DATA  1,4,3,2       :REM ON PERMUTE 2 ET 4
460 DATA        4,3,2,1
470 DATA        3,2,1,4
480 DATA        2,1,4,3
490 DATA  1,6,3,5       :REM 4 PASSE EN DESSOUS
500 DATA        6,3,5,1
510 DATA        3,5,1,6
520 DATA        5,1,6,3
530 DATA  1,5,3,6       :REM ON PERMUTE 5 ET 6
540 DATA        5,3,6,1
550 DATA        3,6,1,5
560 DATA        6,1,5,3
570 DATA  2,5,4,6       :REM METTONS 1 AU-DESSUS
580 DATA        5,4,6,2
590 DATA        4,6,2,5
600 DATA        6,2,5,4
610 DATA  2,6,4,5       :REM METTONS 1 AU-DESSUS
620 DATA        6,4,5,2
630 DATA        4,5,2,6
640 DATA        5,2,6,4
700 WW0
710 FOR L=1 TO 4
720 FOR W=1 TO 4        :REM REMPLISSAGE DE LA TABLE PO
730 READ PO(L,W)
740 NEXT W
750 NEXT L
800 REM ***************************DEBUT*****************************
810 FOR Q1=1 TO 24 STEP 4   :REM ON NE PREND QUE LES 6 POSITIONS DE BASE
820 P(1)=Q1
830 GOSUB 1000              :REM APPEL DE "DEBUT 1"
840 NEXT Q1
850 IF W=0 THEN PRINT "IL N'Y A PAS DE SOLUTION" ELSE PRINT "FIN"
860 END
949 REM ____________________________________________________________
1000 REM1 ************************DEBUT1*****************************
1010 FOR Q2=1 TO 24
1020 P(2)=Q2                        :REM TRANSMISSION DES ARGUMENTS A VERIF
1030 I=1:J=2                        :REM APPEL DE VERIF(C1,C2)
1040 GOSUB 4000                     .REM TEST DU RESULTAT DE VERIF POUR APPEL A DEBUT 2
1050 IF OK=1 THEN GOSUB 2000
1060 NEXT Q2
1070 RETURN                         :REM FIN 1
2000 REM ************************DEBUT 2****************************
2010 FOR Q3=1 TO 24
2020 P(3)=Q3                        .REM TRANSMISSION DES ARGUMENTS DE VERIF
2030 I=1:J=3                        .REM APPEL A VERIF(C1,C3)
2040 GOSUB 4000                     .REM APPEL A VERIF(C2,C3), PUIS A DEBUT 3
2050 IF OK=1 THEN I=2:J=3:
2060 NEXT Q3
2070 RETURN                         :REM FIN 2
3000 REM ************************DEBUT 3****************************
3010 FOR Q4=1 TO 24
3020 P(4)=Q4                        :REM APPEL A VERIF (C1,C4)
3030 I=1:J=4
3040 IF OK<>1 THEN 3170
3050 I=2:J=4: GOSUB 4000
3060 IF OK<>1 THEN 3170
3070 I=3:J=4: GOSUB 4000
3080 IF OK<>1 THEN 3170
3090 W=W+1 PRINT "SOLUTION NUMERO ",W :REM TOUT VA BIEN
3100 PRINT TAB(22)"CUBE 1";TAB(32)"CUBE 2";
3110 PRINT TAB(42)"CUBE 3";TAB(52)"CUBE 4"
3120 FOR K=1 TO 4                   .REM ON IMPRIME LA SOLUTION
3130 PRINT TAB(1)"FACAB ",K, : ",TAB(22)T$(1,(PO(Q1,K))):
3140 PRINT TAB(32)T$(2,(PO(Q2,K))),TAB(42)T$(3,(PO(Q3,K))):
3150 PRINT TAB(52)T$(4,(PO(Q4,K)))
3160 NEXT K
3170 NEXT Q4
3180 RETURN                         :REM FIN 3
3190 REM ************************SOUS PROGRAMME VERIF(C1,C2)*******
4000 REM
4010 OK=1
4020 FOR K=1 TO 4
4030 IF T$(I,(PO(P(I),K)))=T$(J,(PO(P(J),K))) THEN OK=0
4040 REM OK EST MIS A ZERO DANS QU'UNE COULEUR EST REPETEE SUR LA FALAISE
4050 NEXT K
4060 RETURN
```


Figure 6



La procédure VERIF

lage deuxièmement-correct) L'organigramme de cette procédure VERIF est donné en *figure 6*.

Un appel de VERIF appliqué aux cubes Ci et Cj donne la valeur 1 à OK si Ci et Cj sont deuxiemement-corrects, et la valeur 0 sinon.

L'organigramme du programme est représenté par les *figures 7, 8, 9* et *10*. La logique d'ensemble est la suivante : réussir à empiler quatre cubes de façon quatrièmement-correcte, c'est poser le premier cube C1 et pour cette position, essayer toutes les positions troisièmement-correctes des cubes C2, C3, C4. De même empiler les trois cubes C2, C3, C4, c'est empiler C2 sur C1 de façon deuxièmement-correcte et, pour cette position, essayer toutes les positions deuxiemement-correctes de C3 et C4.

On voit ainsi qu'on ne perd pas de temps à empiler les cubes C3 et C4 si les cubes C2 et C1 ne forment pas un empilage deuxièmement-correcte.

Le BASIC n'accepte pas la syntaxe suivante :

FOR P(I) = 1 TO J

il a donc fallu utiliser quatre variables intermediaires, Q1, Q2, Q3 et Q4.

Le sous-programme VERIF a pour argument deux numéros de cubes qui sont transmis avant chaque appel par l'intermédiaire des variables I et J.

Ne vous impatientez pas. Les temps d'exécution sont importants. Le plus court (18 secondes) est pour la configuration du type :
Cube 1 : R, R, R, R, R, R
Cube 2 : R, R, R, R, R, R
Cube 3 : R, R, R, R, R, R
Cube 4 : R, R, R, R, R, R
ou tous les cubes sont de la même couleur : ce qui ne donne bien sûr aucune solution.

On arrive souvent à des temps de l'ordre de 8 minutes avec des configurations du genre :
Cube 1 : R, J, V, B, R, R
Cube 2 : B, R, J, V, R, R
Cube 3 : V, B, R, J, R, R
Cube 4 : J, V, B, R, R, R
qui procure 17 solutions.

Il est possible de « bricoler » le programme. En effet, bien que cela rende le programme moins facile à lire et plus long à écrire, on peut introduire la procedure VERIF dans le corps même du programme. Cela evite d'avoir à executer les instructions du type P(i) = QI des lignes 1020, 2020, 3020 et supprime la transmission d'arguments des lignes 1030, 2030, etc. Nous sommes arrivés à diviser les temps d'exécution par deux. Faites mieux !!! La suite du jeu est bien sûr de trouver une configuration qui ne fournirait qu'une seule solution. C'est un programme assez simple mais qui tournera tres, tres, très longtemps.

On pourrait egalement chercher avec combien de couleurs utilisées, au minimum et au maximum, et suivant quelles dispositions on obtient une seule solution. Les jeux étant ce qu'ils sont, nous vous laissons « jouer aux cubes ».

Jean François Portevin

Figure 7

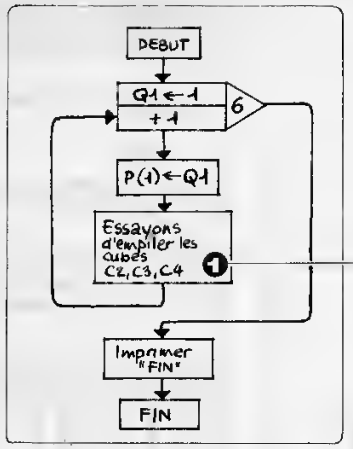
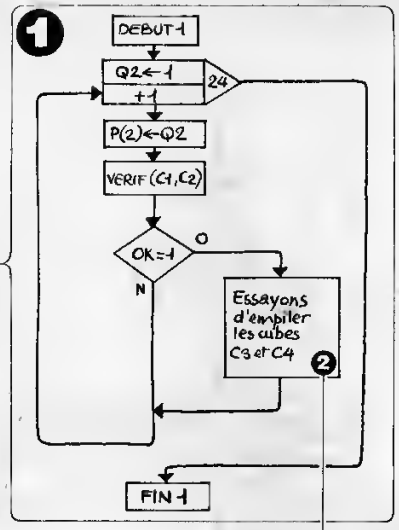
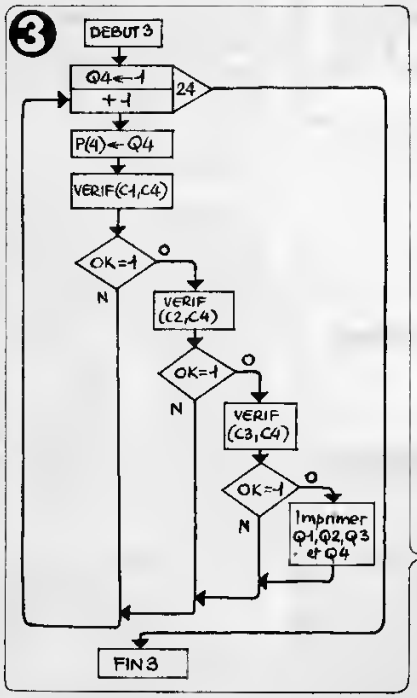
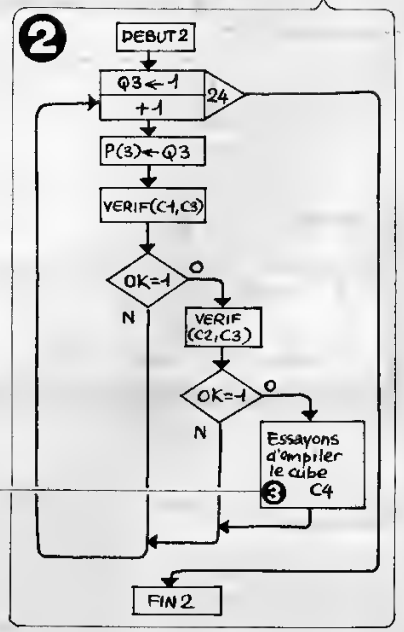


Figure 8







parcourez les méandres d'Isola

avant de défier vos enfants !

Que tous ceux qui en ont assez d'être battus par leur programme d'Othello, se réjouissent, ils vont pouvoir désormais être battus par leur programme d'Isola. Qu'est-ce que Isola ? Isola est un jeu de réflexion qui se joue sur un échiquier 8 × 8 (à l'origine 6 × 8) avec 1 pion pour chacun des 2 joueurs. Le but du jeu est de bloquer le pion ennemi ; pour ce faire vous déplacez à chaque tour votre pion (comme un roi aux échecs) sur une case vide et vous détruisez une case de l'échiquier de votre choix. Si votre adversaire parvient à détruire toutes les cases qui vous entourent, vous ne pouvez plus vous déplacer donc vous avez perdu.

Le programme utilise la technique de la « plongée » pour déterminer la case où il doit se rendre et la case qu'il va détruire sous vos pas

Il compte pour chacune des 8 directions toutes les cases qu'il peut atteindre en 2 coups, ce qui nous donne une triple boucle Vous pouvez d'ailleurs supprimer ou modifier les lignes 2160, 2170 2190 ainsi que 3070, 3080, 3100 pour n avoir qu'une double boucle au cas ou votre BASIC ne serait pas très performant

Le programme joue ensuite la case ayant reçu la meilleure note

La structure du programme est la suivante

Lignes 1040 1060
Initialisation du tableau T terrain de jeu
1 case vide
2 pion de l adversaire
3 pion de la machine
5 case detruite
9 bord du jeu

Lignes 1070-1110
Initialisation du tableau D directions de déplacement

— 11 — 10 — 9
— 1 ↖ + 1
+ 9 + 10 + 11

Initialisation des tableaux P1 et P2 ponderations

Lignes 1120-1170
Initialisation aléatoire des positions des 2 joueurs

Lignes 2040-2090
Recherche de la position des 2 joueurs

Lignes 2120 2230
Recherche de la case à jouer triple boucle A l interieur de chaque boucle on supprime les branches qui font marche arriere ou qui tombent sur une case détruite, au 1er niveau ces directions sont notees —1 dans le tableau de cumul P2 car ce sont des cases interdites

Lignes 2260-2310
On recherche le poste de P2 contenant la valeur la plus elevée, sa position dans la table nous donne la direction a jouer

Lignes 2340-2500
Si la direction à jouer est égale a 0 (aie aie aie) on prend une direction au hasard et on essaye de la jouer Si les 8 directions sont essayees sans succès, on annonce la fin de la partie

Premiere parution L OI n° 22

liste du programme

```
10 REM ********** PROGRAMME ISOLA ***********************************************
20 REM AUTEUR :   PIERRE PALACIOS
30 REM   COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
100 REM*****************************************************************************
110 REM                    DICTIONNAIRE DES VARIABLES
120 REM          A$ =    REPONSE DU JOUEUR
130 REM          C  =    COMPOSANTES VERTICALES DE X
140 REM          D  =    TABLEAU   JEU
150 REM          D0 =    VALEUR DE D3 EN DEBUT DE BOUCLE
160 REM          D2 =    INDICE DE DIRECTION DE LA CASE A DETRUIRE
170 REM          D3 =    INDICE DE DIRECTION A JOUER
180 REM          I2 =    INDICE DE BOUCLE ,1ER NIVEAU
190 REM          I3 =    INDICE DE BOUCLE ,2EME NIVEAU
200 REM          I4 =    INDICE DE BOUCLE ,3EME NIVEAU
210 REM          J  =    INDICE DE BOUCLE AFFICHAGE DU JEU
220 REM          K  =    INDICE DE BOUCLE
230 REM          L  =    COMPOSANTE HORIZONTALE DE X
240 REM          P1 =    TABLEAU   CASE DETRUITE
250 REM          P2 =    TABLEAU   CASE JOUEE
260 REM          T  =    TABLEAU   TERRAIN DE JEU
270 REM          TO =    COMPTEUR DE TOURS
280 REM          U2 =    VALEUR DE LA CASE A DETRUIRE
290 REM          U3 =    VALEUR DU COUP A JOUER
300 REM          V  =    INDICATEUR DE VALIDITE DU COUP DE L'EXPERIMENTATEUR
310 REM          X  =    CASE JOUEE PAR L'EXPERIMENTATEUR
320 REM          X1 =    POSITION DU JOUEUR
330 REM          X2 =    POSITION DE L'ORDINATEUR
1000 REM*****************************************************************************
1010 DIM T(100),D(8),P1(8),P2(8)
1020 REM*****************************************************************************
1030 REM                INITIALISATIONS
1040 FOR K=1 TO 100
1050     READ T(K)
1060 NEXT K
1070 FOR K=1 TO 8
1080     READ D(K)
1090     P1(K)=0
1100     P2(K)=0
1110 NEXT K
1120 X2=INT(RND(0)*80)+11
1130 IF T(X2)>1 THEN 1120
1140 T(X2)=2
1150 X3=INT(RND(0)*80)+11
1160 IF T(X3)>1 THEN 1150
1170 T(X3)=3
1180 TO=0
1190 PRINT"VOULEZ-VOUS JOUER EN PREMIER ?";
1200 GOSUB 3000
1210 IF A$="O" THEN 3500
2000 REM*****************************************************************************
2010 REM                    BOUCLE PRINCIPALE
2020 X2=0
2030 X3=0
2040 FOR K=11 TO 39
2050     IF T(K)<>2 THEN 2070
2060     X2=K
2070     IF T(K)<>3 THEN 2090
2080     X3=K
2090 NEXT K
2100 REM*****************************************************************************
2110 REM                    RECHERCHE DE LA CASE OFFRANT LA PLUS DE LIBERTES POSSIBLES
2120 FOR I2=1 TO 8
2130     IF T(X3+D(I2))>1 THEN 2220
2150     FOR I3=1 TO 8
2150             IF D(I2)=-D(I3) OR T(X3+D(I2)+D(I3))>3 THEN 2200
2160             FOR I4=1 TO 8
2170                     IF D(I3)=-D(I4) OR T(X3+D(I2)+D(I3)+D(I4))>1 THEN 2190
2180                     P2(I2)=P2(I2)+1
2190             NEXT I4
2200     NEXT I3
2210     GOTO 2230
2220     P2(I2)=-1
2230 NEXT I2
2240 D3=0
2250 V3=0
2260 FOR K=1 TO 8
2270     IF P2(K)<V3 THEN 2300
2280     D3=K
2290     V3=P2(K)
2300     P2(K)=0
2310 NEXT K
2320 IF V3>5 THEN 2360
2330 PRINT" JE SENS MA FIN PROCHE !!!"
2340 IF D3=0 THEN 2470
2350 T(D(D3)+X3)=3
2360 T(X3)=1
2370 GOTO 3010
2380 D3=D3+1
2390 IF D3<>8 THEN 2410
2400 D3=D3-8
2410 IF D3<>D0 THEN 2490
2420 PRINT" J'AI PERDU "
2430 PRINT "EN ",TO," COUPS !!"
2440 END
2450 REM*****************************************************************************
2460 REM                    QUAND IL N'Y A RIEN DE BON A JOUER ON CHOISIT AU HASARD
2470 D3=INT(RND(0)*8+1)
2480 D0=D3
2490 IF T(D(D3)+X3)>1 THEN 2380
2500 GOTO 2350
3000 REM*****************************************************************************
3010 REM                    DESTRUCTION DE LA CASE OFFRANT LE PLUS DE LIBERTES A L'ENNEMI
3020 FOR I2=1 TO 8
3030     IF T(X2+D(I2))>1 THEN 3130
3040     FOR I3=1 TO 8
3050             IF D(I3)=-D(I2) THEN 3110
3060             IF T(X2+D(I2)+D(I3))>1 THEN 3110
3070             FOR I4=1 TO 8
3080                     IF D(I4)=-D(I3) OR T(X2+D(I2)+D(I3)+D(I4))>1 THEN 3100
3090                     P1(I2)=P1(I2)+1
3100             NEXT I4
3110     NEXT I3
3120     GOTO 3140
3130     P1(I2)=-1
3140 NEXT I2
3150 D2=0
3160 V2=0
3170 FOR K=1 TO 8
3180     IF P1(K)<V2 THEN 3220
3190     D2=K
3200     V2=P1(K)
3210     P1(K)=0
3220 NEXT K
3230 IF D2<>0 THEN 3300
3240 REM  SI L'ENNEMI N'A PRESQUE PLUS DE LIBERTES ON CHOISIT
3250 REM LA PREMIERE CASE QUI SE PRESENTE(OK TOUTES FACONS IL A PERDU)
3260 FOR K=1 TO 8
3270     IF T(X2+D(K))>1 THEN 3290
3280     D2=K
3290 NEXT K
3300 T(X2+D(D2))=5
3310 IF V2>5 THEN 3570
3320 PRINT" JE SENS VOTRE FIN PROCHE!!!"
3500 REM*****************************************************************************
3510 REM                    AFFICHAGE DU JEU
3520 PRI  CONTROL L=EFFACEMENT PAGE
3530 REM  CONTROL J=DESCENTE D'UNE LIGNE
3540 REM  CONTROL C=BEEP!!!
3550 REM  CONTROL I=TABULATION
3560 REM*****************************************************************************
3570 PRINT"     1  2  3  4  5  6  7  8"
3580 FOR K=0 TO 9
3590     PRINT",K," ",
3600     FOR J=1 TO 10
3610             IF T(K*10+J)<>9 THEN 3630
3620             PRINT"###",
3630             IF T(K*10+J)<>5 THEN 3650
3640             PRINT"###",
3650             IF T(K*10+J)<>2 THEN 3670
3660             PRINT" 1 ",
3670             IF T(K*10+J)<>3 THEN 3690
3680             PRINT" 2 ",
3690             IF T(K*10+J)<>1 THEN 3710
3700             PRINT" . ",
3710     NEXT J
3720     PRINT",K
3730 NEXT K
3740 PRINT"     1  2  3  4  5  6  7  8"
4000 REM*****************************************************************************
4010 REM                    ON CHERCHE A VOIR SI L'ON A GAGNE
4020 D2=0
4030 X=0
4040 FOR K=1 TO 8
4050     IF T(D(K)+X2)>1 THEN 4080
4060     D2=D2+1
4070     X=D(K)+X2
4080 NEXT K
4090 IF D2>0 THEN 5020
4100 PRINT"    J'AI    GAGNE EN ",TO," COUPS!!!"
4110 END
```

Liste du programme (suite)

```
5000 REM*****************************************************************************
5010 REM                    JEU DE L'EXPERIMENTATEUR
5020 IF D2=1 THEN 5060
5030 PRINT"PARMI LES ",D2," COUPS JOUABLES,LEQUEL JOUEZ-VOUS ?"
5040 PRINT"LIGNE PUIS COLONNE";
5050 GOTO 5110
5060 X=X+1
5070 L=INT(X/10)
5080 C=X-L*10
5090 PRINT"VOUS NE POUVEZ PLUS JOUER QUE ",D," + ",D," !!!"
5100 GOTO 5130
5110 INPUT L,C
5120 IF C<1 OR C>8 OR L<1 OR L>8 THEN 5090
5130 X=10*L+C+1
5140 IF T(X)<>1 THEN 5090
5150 D0=X-X2
5160 IF D0>11 OR D0<-11 THEN 5090
5170 V=0
5180 FOR K=1 TO 8
5190     REM   SI VOUS NE DISPOSEZ PAS DE CES TESTS MODIFIEZ DONC
5200     REM   L'INSTRUCTION CI-DESSOUS A LA PLACE DE LA LIGNE 1650
5210     REM   IF D0=D(K) THEN V=V+1
5220     V=V+1*(D0=D(K))
5230 NEXT K
5240 IF V=0 THEN 5090
5250 T(X)=2
5260 T(X2)=1
5270 TO=TO+1
5500 REM*****************************************************************************
5510 REM                    DESTRUCTION DE LA CASE
5520 PRINT"QUELLE CASE DETRUISEZ-VOUS ?";
5530 INPUT L,C
5540 IF C<1 OR C>8 OR L<1 OR L>8 THEN 5520
5550 X=L*10+C+1
5560 IF T(X)>1 THEN 5520
5570 T(X)=5
5580 GOTO 2010
9000 REM*****************************************************************************
9010 REM                    SOUS PROGRAMME POUR OBTENIR L'UNE DES REPONSES "O" OU "N"
9020 INPUT A$
9030 IF A$<>"O" AND A$<>"N" THEN PRINT "O OU N", GOTO 3010
9040 RETURN
10000 REM****************************************************************************
10010 REM  I
10020 DATA 9,9,9,9,9,9,9,9,9,9
10030 DATA 9,1,1,1,1,1,1,1,1,9
10040 DATA 9,1,1,1,1,1,1,1,1,9
10050 DATA 9,1,1,1,1,1,1,1,1,9
10060 DATA 9,1,1,1,1,1,1,1,1,9
10070 DATA 9,1,1,1,1,1,1,1,1,9
10080 DATA 9,1,1,1,1,1,1,1,1,9
10090 DATA 9,1,1,1,1,1,1,1,1,9
10100 DATA 9,1,1,1,1,1,1,1,1,9
10110 DATA 9,9,9,9,9,9,9,9,9,9
10120 REM  D
10130 DATA -1,9,-11,-10,1,-9,11,10
```

Exemple simple d'exécution

Voulez vous jouer en premier ? OUI

```
   1 2 3 4 5 6 7 8
0 ##################### 0
1 ##.......... ##  1
2 ##.......... ##  2
3 ##.1........ ##  3
4 ##.......... ##  4
5 ##.......... ##  5
6 ##......2... ##  6
7 ##.......... ##  7
8 ##.......... ##  8
9 ##################### 9
   1 2 3 4 5 6 7 8
```

Parmi les 8 coups jouables lequel jouez-vous ?
Ligne puis colonne 4,3
Quelle case detruisez vous ? 6,5

```
   1 2 3 4 5 6 7 8
0 ##################### 0
1 ##.......... ##  1
2 ##.......... ##  2
3 ##...##..... ##  3
4 ##..1....... ##  4
5 ##.....2.... ##  5
6 ##....##.... ##  6
7 ##.......... ##  7
8 ##.......... ##  8
9 ##################### 9
   1 2 3 4 5 6 7 8
```

Parmi les 7 coups jouables, lequel jouez-vous ?
Ligne puis colonne 4,4
Quelle case detruisez-vous ? 5,5

```
   1 2 3 4 5 6 7 8
0 ##################### 0
1 ##.......... ##  1
2 ##.......... ##  2
3 ##...##..... ##  3
4 ##...1...2... ##  4
5 ##...##### ... ##  5
6 ##....##.... ##  6
7 ##.......... ##  7
8 ##.......... ##  8
9 ##################### 9
   1 2 3 4 5 6 7 8
```

Parmi les 5 coups jouables, lequel jouez-vous ?
Ligne puis colonne 3,5
Quelle case detruisez-vous ? 5,7

```
   1 2 3 4 5 6 7 8
8 ##################### 8
1 ##.......... ##  1
2 ##...##..... ##  2
3 ##...##1 2... ##  3
4 ##.......... ##  4
5 ##...##### .## . ##  5
6 ##....##.... ##  6
7 ##.......... ##  7
8 ##.......... ##  8
9 ##################### 9
   1 2 3 4 5 6 7 0
```

Parmi les 5 coups jouables, lequel jouez-vous ?

Ligne puis colonne 4,5

Quelle case detruisez-vous ? 2,5

Quelques coups plus tard

Je sens votre fin proche !!!

```
   1 2 3 4 5 6 7 6
6 ##################### 6
1 ##....##.##.## 1
2 ##...##### 1 ##.## 2
3 ##...##.######### 3
4 ##...##.######### 4
5 ##..#######.##.## 5
6 ##....#####..## 6
7 ##.....##2.## 7
8 ##.......## 8
9 ##################### 9
   1 2 3 4 5 6 7 8
```

Parmi les 2 coups jouables, lequel jouez-vous ?
Ligne puis colonne 1,6
Quelle case detruisez vous ? 8,6
Je sens votre fin proche !!!

```
   1 2 3 4 5 6 7 8
8 ##################### 8
1 ##....##1.## 1
2 ##...######### 2
3 ##...##.######### 3
4 ##...##.######### 4
5 ##..#######.##.## 5
6 ##....#####.2## 6
7 ##.....##..## 7
8 ##.....##..## 8
9 ##################### 9
   1 2 3 4 5 6 7 8
```

J AI GAGNE EN 11 COUPS !

Lignes 3020-3320

On refait les mêmes calculs afin de déterminer la case à supprimer autour de l'ennemi — voir les lignes 2120 à 2500. On utilise X2 et P1 au lieu de X3 et P2.

Lignes 3500 à 3740

Impression du tableau de jeu. (Il reste quelque chose à faire quand K vaut 0 ou 9 aux lignes 3590 et 3720).

Lignes 4000 à 4080

Recherche du nombre de cases jouables par l'adversaire. Si ce nombre est égal à 0 on annonce la victoire avec force controls G, si le nombre est égal à 1, on joue le coup à la place du joueur, sinon on lui demande la case ou veut jouer.

Lignes 5110-5580

Entrée et contrôle du coup du joueur, d'abord la case où il se rend (voisine de la case de départ et libre), puis la case qu'il détruit (cette case ne doit pas déjà avoir été détruite).

On recommence alors au début jusqu'à ce que mort s ensuive (sic !)

Les lignes 3000 à 3320 peuvent être transformées en sous-programme (appelé 2 fois par coup de la machine) en vue de supprimer les lignes 2120 à 2500, mais il sera alors plus difficile de faire évoluer la technique de jeu du programme.

La partie peut durer 32 coups au maximum à raison de 2 cases détruites par coup, mais généralement la partie se termine au bout de 13 à 18 coups. Si elle se termine avant cela veut dire que vous êtes soit très bon, soit très mauvais.

Il ne vous reste plus maintenant qu'à essayer Isola et lorsque vous aurez acquis une dextérité telle que vous gagnez à tous les coups (n'est-ce pas ?), pourquoi ne pas compliquer un peu le jeu en définissant un damier de 15 cases sur 15, un déplacement de 1 à 3 cases, la possibilité de détruire entre 1 et 5 cases à chaque tour de jeu ou toute autre convention. Ce jeu simple a priori pourra alors se révéler redoutable surtout si vous faites intervenir un soupçon de hasard pour corser le tout.

Pierre Palacios

Note : la mise au point de ce programme a été effectuée à l'A.R.T.A. au centre Georges Pompidou. Il a été réalisé à partir de l'idée présentée dans le jeu Isola des Editions Ravensburger, jeu qui utilise un damier 6 × 8.

pensez-vous vraiment casser des briques ?

Connaissez-vous le « casse-briques » ? Il s'agit d'un jeu, dit jeu vidéo, car il associe des déplacements sur l'écran et une action réflexe de votre part. Ce n'est pas un jeu très difficile mais vous verrez bien vite, hélas, que vos enfants s'y montrent rapidement plus habiles que vous. Pour en avoir la démonstration, il vous suffit d'entrer la liste du programme en BASIC sur votre PET ou votre CBM.

Le « mur de briques » n'est pas un jeu de réflexion, mais un jeu de réflexe. Ne le dédaignez pas pour autant : il nécessite une bonne coordination entre acuité visuelle et, en l'occurrence, déplacement de la main.

Le but du jeu est simple : en haut de l'écran est affiché un mur de briques, en bas une raquette, style raquette de tennis, stylisée bien sûr ! Entre les deux, une petite balle se déplace. Chaque fois qu'elle heurte une brique, la brique disparaît et vous marquez un point. Vers le bas, soit vous la touchez avec votre raquette et elle repart vers le haut, soit vous la manquez et vous avez perdu.

Vous avez perdu pour cette balle tout au plus car une nouvelle apparaît mais vous avez tout de même perdu du temps et c'est contre le temps que vous jouez : il vous faut détruire toutes les briques en un minimum de temps.

Le déplacement de la raquette s'effectue à l'aide des touches 4 – pour aller à gauche – et 6 pour aller à droite.

Comme le jeu ne se veut pas trop difficile, vous ne sortirez par le bas de l'écran qu'une fois sur deux.

Il vous sera possible d'utiliser ou de supprimer le son (les listes du programme et des variables vous indiquent les lignes et variables affectées).

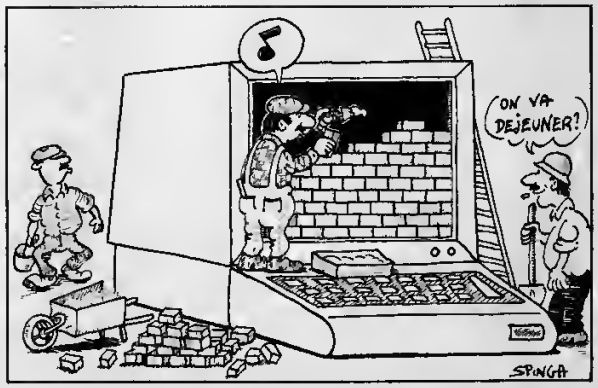
L'adaptation à un PET ou à un CBM consiste simplement à modifier la variable KB (voir les lignes 1300 et 1310) KB = 515 correspond à un PET et KB = 151 à un CBM.

Ce programme utilise bien sûr divers POKEs et il est propre aux appareils PET/CBM.

Il ne vous reste plus qu'à vous entraîner et à essayer de défier ensuite vos enfants. Peut-être n'est-ce pas perdu d'avance ?

Jean-Pierre Jaborska

```
          * VARIABLES *
SN  . REGISTRE 59464 : SON
MU  . 208 = BRIQUE
SP  . 32  = ESPACE
AO  : 214 = BORD LATERAL
BO  : 224 = INVERSE ESPACE
FO  : 192 = BORD BAS
SO  : 111 = SORTIE BALLE
BA  : 81  = BALLE
SC  . 32768 = DEPART ECRAN
KB  . PET = 515 / CBM =151
      MEMOIRE DU CODE DE TOUCHE PRESSEE
A   . VARIABLE INTERMEDIAIRE
NB  . NOMBRE DE BALLES
B   . COMPTEUR DE BOUCLE
DI  : DIRECTION BALLE
PA  . POSITION DE LA RAQUETTE
CI  : 160 = RAQUETTE
PO  : POSITION DE LA BALLE
SI  : SIGNE DIRECTION BALLE
PV  . POSITION BALLE A EFFACER
A$  . CARACTERE ENTRE PAR LE CLAVIER
PI  . POINTS JOUEUR
RA  . VARIABLE INTERMEDIAIRE
T   : SON SORTIE BALLE
JK  : SON SORTIE BALLE
TT  : DRAPEAU
FF  : DRAPEAU
```



Première parution : L'OI n° 32

Liste du programme

```
READY.
 10 REM        * MUR DE BRIQUES *
 20 REM      * MIS AU POINT SUR CBM (3,9 K) *
 30 REM  AUTEUR : JEAN PIERRE JABORSKA
 40 REM  COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
 50 REM*****************************************************
 100 GOSUB 1000        : REM MODE D'EMPLOI
 110 GOSUB 1200       : REM INITIALISATION
 120 GOSUB 1400       : REM DESSIN DU TERRAIN
 130 GOSUB 1800       : REM BALLE
 140 GOSUB 1900       : REM DEBUT DU JEU
 150 IF FF=1 THEN 210
 160 IF TT=1 THEN GOSUB 2300: REM UNE BALLE EN MOINS
 170 IF NB<20 AND PI<100 THEN 130
 180 GOSUB 2500       : REM FIN DE PARTIE
 190 GOSUB 9000       : REM TEST SUR LA REPONSE
 200 IF A$="O" THEN RUN 110
 210 POKE SN,0
 220 POKE SN+2,0
 230 POKE SN+3,0
 240 PRINT""
 250 CLR
 260 END
 1000 REM*****************************************************
 1010 REM          PRESENTATION DU JEU
 1020 PRINT""
 1030 PRINT:PRINT
 1040 PRINT"MUR DE BRIQUES"
 1050 PRINT: PRINT
 1060 PRINT"BUT DU JEU : DETRUIRE TOUTES LES"
 1070 PRINT"        BRIQUES,A L'AIDE D'UNE BALLE,"
 1080 PRINT"        EN UN MINIMUM DE TEMPS."
 1090 PRINT: PRINT
 1100 PRINT"POUR CELA ON DEPLACE UNE RAQUETTE"
 1110 PRINT"EN APPUYANT SUR LES TOUCHES:"
 1120 PRINT"    4 POUR ALLER A GAUCHE"
 1130 PRINT"    6 POUR ALLER A DROITE"
 1140 PRINT: PRINT"POUR ARRETER LE JEU EN COURS"
 1150 PRINT"DE PARTIE APPUYER SUR F"
 1160 PRINT: PRINT"      ETES-VOUS PRET";
 1170 GOSUB 9000
 1180 IF A$="N" THEN 1020
 1190 RETURN
 1200 REM-----------------------------------------------
 1210 REM          INITIALISATION
 1220 PRINT""
 1230 POKE 59468,12
 1240 SN=59464: POKE SN+3,16: REM REGISTRE SON PAR CB2
 1250 POKE SN+2,81: POKE SN,0
 1260 MU=206: SP=32: REM CARACTERE
 1270 AO=214: BO=224
 1280 FO=192: BA=81
 1290 SO=111: CI=160
 1300 SC=32768: KB=515
 1310 IF PEEK(50000) THEN KB=151: REM TYPE DE PET
 1320 RETURN
 1400 REM-----------------------------------------------
 1410 REM          PREPARE L'ECRAN
 1420 REM--------
 1430 FOR A=0 TO 79
 1440 POKE SC+A,BO
 1450 NEXT A
 1460 REM--------
 1470 REM--------
 1480 FOR A=2 TO 23
 1490 POKE SC+(A*40),AO
 1500 POKE SC+(A*40)+39,AO
 1510 NEXT A
 1520 REM--------
 1530 REM--------
 1540 FOR A=0 TO 39
 1550 POKE SC+(24*40)+A,FO
 1560 NEXT A
 1570 REM--------
 1580 REM--------
 1590 FOR A=3 TO 36 STEP 2
 1600 POKE SC+(24*40)+A,SO
 1610 NEXT A
 1620 REM--------
 1630 PRINT""TAB(5)"POINTS"TAB(20)"BALLE NO : 1"
 1640 NB=1
 1650 REM--------
 1660 FOR B=2 TO 5
 1670 REM------
 1680 FOR A=1 TO 38
 1690 POKE SC+(40*B)+A,MU
 1700 NEXT A
 1710 REM------
 1720 NEXT B
 1730 REM--------
 1740 DI=40
 1750 TI$="000000"
 1760 PA=33627: REM DEPART RAQUETTE
 1770 POKE PA+1,CI
 1780 POKE PA,CI: POKE PA+1,CI
 1790 RETURN
 1800 REM-----------------------------------------------
 1810 REM          DEPART DE LA BALLE
 1820 PO=33268
 1830 SI=1: POKE PO,BA
 1840 RETURN
 1900 REM-----------------------------------------------
 1910 REM          LE JEU COMMENCE ICI
 1920 A=PEEK(KB)
 1930 IF A>42 OR A<41 THEN 2010
 1940 PV=PA
 1950 PA=PA+(A=42)*2-(A=41)*2
 1960 IF PA<33610 THEN PA=33610: PV=PA+2
 1970 IF PA>33643 THEN PA=33643: PV=PA-2
 1980 POKE PV+2,SP: POKE PV+1,SP
 1990 POKE PV,SP: POKE PA,CI
 2000 POKE PA+1,CI: POKE PA+2,CI
 2010 GET A$
 2020 IF A$="F" THEN FF=1: GOTO 2220
 2030 A=PEEK(PO+DI)
 2040 IF A=BO OR A=CI OR A=FO THEN SI=-SI: GOTO 2160
 2050 IF A=AO THEN POKE SN,140
 2060 IF A<>MU THEN 2100
 2070 SI=-SI: POKE (PO+DI),SP
 2080 DI=ABS(DI): POKE SN,99
 2090 PI=PI+1: PRINT"";PI
 2100 IF A<>AO AND A<>MU THEN 2120
 2110 DI=DI+((40+DI)*2*(DI<0))-((40-DI)*2*(DI>0))
 2120 IF A=SO THEN TT=1: GOTO 2220
 2130 IF PEEK(PO)=BA THEN POKE PO,SP
 2140 PO=PO+DI: POKE PO,BA
 2150 POKE SN,0: GOTO 1920
 2160 RA=INT((RND(1)*21+10))/10
 2170 DI=(38+RA)*SI
 2180 IF A=FO THEN POKE SN,65
 2190 IF A=BO THEN POKE SN,140
 2200 IF A=CI THEN POKE SN,220
 2210 GOTO 1920
 2220 RETURN
 2300 REM-----------------------------------------------
 2310 REM          SORTIE BALLE
 2320 POKE PO,SP: POKE PO+DI,BA
 2330 POKE SN+2,51
 2340 REM--------
 2350 FOR T=1 TO 3
 2360 REM------
 2370 FOR JK=32*T TO 80*T STEP 3
 2380 POKE SN,JK
 2390 NEXT JK
 2400 REM------
 2410 NEXT T
 2420 REM--------
 2430 POKE SN+2,81:POKE SN,0
 2440 NB=NB+1
 2450 PRINT"";SPC(30)NB
 2460 POKE PO+DI,SO
 2470 RETURN
 2500 REM-----------------------------------------------
 2510 REM          FIN DE PARTIE
 2520 PRINT"PARTIE TERMINEE APRES"
 2530 PRINT""MID$(TI$,3,2);
 2540 PRINT" MINUTES";RIGHT$(TI$,2);" SECONDES"
 2550 PRINT"VOULEZ VOUS JOUER UNE"
 2560 PRINT"AUTRE PARTIE   ";
 2570 RETURN
 9000 REM-----------------------------------------------
 9010 REM          TEST SUR LA REPONSE
 9020 INPUT A$
 9030 A$=LEFT$(A$,1)
 9040 IF A$<>"O" AND A$<>"N" THEN 9020
 9050 RETURN
READY.
```


partez à l'aventure dans le pays fantastique des Donjons et des Dragons

Connaissez-vous les jeux de rôles ? Non. Alors suivez-nous à travers la lande aride vers le château du Dragon. Attention, le voyage sera périlleux, semé d'embûches et surtout peuplé de divers monstres qui assurément ne vous veulent pas du bien. Avancez lentement, j'entends déjà les premiers hurlements qui montent de la nuit...

Depuis des siecles le chateau etait proche du village Depuis des siecles le tresor etait dans le château Et depuis des siecles, des hommes inconscients tentaient de ravir le tresor

Il y avait eu Hérold heritier d un roi Elfe armé de son arc flamboyant Il avait peri de la main d un Troll Il y avait eu Lovin vaillant combattant des guerres elfiques Sa hache n avait pu venir a bout d un Balrog

Des hommes des nains des hobbits avaient tenté d affronter le chateau Les rares qui revinrent avaient laisse leur esprit dans la lande hostile Les pieges du château du Dragon sont de ceux auxquels nul ne resiste

Certains parmi les plus vieux affirment que les restes des precedents explorateurs defendent eux-memes le tresor dont ils n ont pu s emparer La nuit des cris aux intonations rien moins qu humaines, parviennent jusqu aux maisons du village Nul ne pourrait dire avec exactitude ce qui se passe la bas mais l imagination supplante la connaissance et les villageois décrivent a ceux qui veulent ecouter les monstres hantant le chateau

Il y a le Troll enorme dont le corps semble fait de pierres l or-

que barde de fer et dont les crocs scintillent sous la lune le centaure rapide et puissant le Balrog, massif et aile la goule et le vampire assoiffes de sang le gobelin dont la cruauté n a d égal que la bêtise , et bien d autres

Et puis il y a celui dont le nom fait s enfuir les plus courageux le maître des lieux le Dragon Aux aguets dans sa taniere il attend l imprudent qui osera le defier

Jadis (c etait avant les guerres elfiques) Saroumane le Magnifique demanda l hospitalite aux habitants du château Ceux-ci, farouches partisans des Elfes la lui refuserent Et l on dit que Saroumane furieux de ce refus transforma le châtelain en dragon et les autres habitants en divers types de monstres Puis il amena le trésor qu il tenait de ses divers pillages, dans le château et condamna le dragon a le proteger pour l eternite

Telle est la legende que racontent les Elfes le soir auprès du feu mais les hobbits ont une autre version de

(suite du texte p 55)

```
10 REM ***********************************************
20 REM  CONTRAT DE DONJON ET DRAGONS
30 REM  ********************************************
40 REM  ********************************************
100 CLS
110 CLEAR 1000
120 DIM S(20),A(29),B(33),P(30),Q(20),A$(50)
130 PRINT  PRINT  PRINT
140 PRINT"              LE CONTRAT DU DRAGON"
150 PRINT  PRINT  PRINT
160 PRINT"    IF VOUS ETES PRET, UNE FOIS APPUYES SUR "E" E"
170 PRINT"    ""E""""""""""
180 GOSUB 5000  GOSUB 5600
190 A=0  Y=0
200 GOSUB 1200                    REM I INITIALISATIONS DES MONSTRES
210 GOSUB 1300                    REM CARACTERISTIQUES DU PERSONNAGE
220 GOSUB 1400                    REM PLAN DES LIEUX
230 FOR I=3 TO 6: PRINT@64*I+5" "  : NEXT I
240 GOSUB 1600                    REM DEPLACEMENTS
250 REM  DETERMINATION DE LA SUITE ATTENTE
260 IF X<20 AND Y<13 THEN GOSUB 7200: GOTO 270          REM VILLAGE
270 IF (X<30 AND X>23) AND (Y>29 AND Y>34) THEN 350
280 IF X>32 AND X>24 THEN GOSUB 3000: GOSUB 320         REM CHATEAU
290 IF Z=4 THEN 350
300 IF Z=2 OR Z=0 THEN 230
310 IF Z=3 THEN 380
320 IF X>42 AND Y>18 THEN GOSUB 3600: GOTO 340          REM ARBRES DU CHATEAU
330 GOSUB 4000                                          REM LANDE VIDE
340 IF A>0 THEN 230 ELSE 380
350 GOSUB 2400                                          REM SALLE DU TRESOR
360 IF A>6 THEN 380
370 IF A4=0 OR Q=4 THEN 230
380 END
1000 REM*********************************************
1010 REM                    PLAN DES LIEUX
1020 CLS
1030 PRINT                  PLAN DES LIEUX DU CHATEAU DU DRAGON"
1040 PRINT ----------------------------------------------:"
1050 PRINT"I                    I                NORD                    I:"
1060 PRINT"I VILLAGE I                                                   I:"
1070 PRINT"I----------------                LANDE                        I:"
1080 PRINT"I OUEST                                                    EST I:"
1090 PRINT"I                                                             I:"
1100 PRINT"I                    I      ARBRES DU CHATEAU                 I:"
1110 PRINT"I                    I                                        I:"
1120 PRINT"I                    I                                        I:<ENTREE
1130 PRINT"I                    I                                        I:"
1140 PRINT"I                    I                                        I:"
1150 PRINT"I                    I                                        I:"
1160 PRINT"I                    I                                        I:"
1170 PRINT"I                    I                                        I:"
1180 PRINT"I                    I                                        I:"
1190 PRINT"I                    I                                        I:"
1200 PRINT"I                    I                                        I:"
1210 PRINT"I                    I                                        I:"
1220 PRINT"I                    I                                        I:"
```

```
3250 A1=A1+R(5)*5
3260 GOSUB 5000
3270 ZZ=2
3280 RETURN
3300 REM==================================================
3310 REM                 OUBLIETTES DU CHATEAU
3320 GOSUB 5900
3330 CLS
3340 PRINT"AAAAAAAAAAA !!"
3350 PRINT"VOUS ETES TOMBE DANS LES OUBLIETTES ."
3360 GOSUB 5400
3370 ZZ=3
3380 RETURN
3400 REM==================================================
3410 REM                 PIEGE DU CHATEAU
3420 B$(3)= VOUS AVEZ OCCASIONNE UN PIEGE.                 "
3430 B$(4)="UNE PLANTE VIENT DE SE PLANTER DANS VOTRE BRAS "
3440 GOSUB 5900
3450 A1=A1-RND(3)
3460 GOSUB 5000
3470 IF A1<=0 THEN ZZ=3  CLS  GOSUB 5400  GOTO 3490
3480 ZZ=2
3490 RETURN
3500 REM==================================================
3510 REM                 PIEGES DU CHATEAU
3520 B$(3)= VOUS PAR UNE PROVISION DE LA CHUTE DE PIERRES "
3530 B$(4)="VOUS ETES BLESSE                              "
3540 GOSUB 5900
3550 A1=A1-RND(6)
3560 GOSUB 5000
3570 IF A1<=0 THEN ZZ=3  CLS  GOSUB 5400  GOTO 3590
3580 ZZ=2
3590 RETURN
3600 REM==================================================
3610 REM                 ARCADES DU CHATEAU
3620 D=0  D1=0  D2=0  D4=0
3630 TT=0
3640 IF D3=1 AND D5=1 THEN TT=3  GOTO 3690
3650 IF D3=1 THEN TT=1  GOTO 3690
3660 B$(0)="VOUS ETES PRES DU CHATEAU. LES MONSTRES POLLUENT PAR ICI"
3670 D3=1
3680 IF T1<>0 AND D5=0 THEN B$(1)="
                                  B$(2)=B$(1)  D5=1  GOTO 3720
3690 B$(1)="VOUS POUVEZ, SI VOUS LE DESIREZ, ALLER REPAIRE VOS FORCES AU "
3700 B$(2)=  VILLAGE ( A VOS RISQUES ET PERILS )                          "
3710 Z1=1
3720 T=9  GOSUB 4300
3730 B$(5)=""
3740 RETURN
3800 REM==================================================
3810 REM                 LANDE ARIDE
3820 D=0  D1=0  D2=0  D3=0
3830 TT=0
3840 IF D4=1 THEN TT=3  GOTO 3890
3850 B$(0)= VOUS ETES DANS LA LANDE ARIDE QUI SEPARE LE VILLAGE DU CHATEAU "
3860 B$(1)="VOUS POUVEZ Y FAIRE DE MAUVAISES RENCONTRES                    "
3870 B$(2)="    (EN EFFET ...)                                             "
3880 D4=1
3890 T=11  GOSUB 4000
3900 B$(5)=""
3910 RETURN
4000 REM==================================================
4010 REM          COMBAT (OU NON) AVEC LE MONSTRE
4020 IF RND(12)<T THEN B$(3)="TOUT A L'AIR TRANQUILLE, POURSUIVEZ VOTRE
                              CHEMIN                                  "
                              GOSUB 5900  GOTO 4060
4030 B$(3)="DE SENS QU'IL VA SE PASSER QUELQUE CHOSE !!!"
4040 M=RND(100)  RESTORE
4050 FOR I=1 TO U
4060   READ S
4070 NEXT I
4080 IF S>M THEN GOSUB 4700  GOSUB 5900  GOTO 4050
4090 GOSUB 5900
4100 GOSUB 5600
4110 CLS
4120 S=S-3
4130 M=M+1
4140 A$(M)=M$(S)
4150 PRINT GS(RND(10)),  !!!      ",M$(S)
4160 M1=H(S)
4170 M2=Q(S)
4180 M3=F(S)
4190 M4=G(S)
4200 PRINT"VOUS DEVEZ COMBATTRE"
4210 PRINT"                POINTS DE VIE     FORCE DE FRAPPE    RESISTANCE"
4220 PRINT"    VOUS        ",A1,"                I A S +",A2,"              ",A3
4230 PRINT TAB(2) M$(S);
4240 PRINT TAB(16) M1,
4250 PRINT TAB(32)"1 A"M2"+",M3
4260 PRINT TAB(53) M4
4270 GOSUB 5600  GOSUB 5600
4280 PRINT"LE COMBAT
4290 IF RND(2)=1 THEN PRINT"VOUS FRAPPEZ LE PREMIER "  GOTO 4450
4300 PRINT"LA MONSTRE FRAPPE EN PREMIER "
4310 GOSUB 5600
4320 IF RND(10)<A3 THEN PRINT"LE MONSTRE RATE SON COUP !!!   A VOUS "
                                                        GOTO 4450
4330 PRINT"LE MONSTRE VOUS A TOUCHE "
4340 GOSUB 5600
4350 B=RND(M2)+M3
4360 GOSUB 4900
4370 A1=A1-B
4380 GOSUB 5600
4390 IF A1<=0 THEN GOSUB 5300  GOTO 5550
4400 PRINT"VOTRE NOUVEAU TOTAL : ",A1," POINT",
4410 IF A1<>1 THEN PRINT"S",
4420 PRINT" DE VIE "
4430 GOSUB 5600
4440 PRINT"A VOUS DE FRAPPER "
4450 GOSUB 5600
4460 IF RND(10)<A4 THEN PRINT"VOUS RATEZ VOTRE COUP !!!   AU MONSTRE."
                                                        GOTO 4510
4470 PRINT"VOUS AVEZ TOUCHE LE MONSTRE "
4480 GOSUB 5600
4490 B=RND(5)+2
4500 GOSUB 4900
4510 M1=M1-B
4520 GOSUB 5600
4530 IF M1<=0 THEN 4580
4540 PRINT"LE NOUVEAU TOTAL DU MONSTRE  ";M1;"POINT";
4550 IF M1<>1 THEN PRINT"S"
4560 PRINT" DE VIE."
4570 PRINT" AU MONSTRE DE FRAPPER !"  GOTO 4310
4580 GOSUB 5600
4590 PRINT"APRES CE COMBAT ACHARNE, VOUS AVEZ VAINCU ";M$(S)"."
4600 PRINT" BON POUR LE MORAL  MAIS CE N'EST PAS ENCORE TERMINE ..."
4610 GOSUB 5600
4620 PRINT"VOTRE TOTAL ACTUEL  ";A1;"POINT"
4630 IF A1<>1 THEN PRINT"S DE VIE." ELSE PRINT "DE VIE."
4640 GOSUB 5800  GOSUB 5600
4650 GOSUB 1000
4660 RETURN
4700 REM==================================================
4710 REM                 TROUVAILLES !
4720 ON R GOTO 4730,4770,4810
4730 B$(4)="VOUS AVEZ TROUVE UNE BOUTEILLE D'ELIXIR.                      "
4740 M$(S)="EN LA BUVANT  VOTRE VITALITE AUGMENTE                         "
4750 A1=A1+RND(6)
4760 GOSUB 5900  GOTO 4850
4770 B$(4)="VOUS AVEZ TROUVE UNE ARME MAGIQUE QUI AUGMENTE VOTRE FORCE    "
4780 A2=A2+1
4790 IF A2>3 THEN A2=3
4800 GOTO 4840
4810 B$(4)= VOUS AVEZ TROUVE UNE ARMURE QUI VOUS PROTEGE
4820 A3=A3+1
4830 IF A3>3 THEN A3=3
4840 RETURN
4900 REM==================================================
4910 REM                 COMMENTAIRES DU COMBAT
4920 IF B>8 THEN PRINT"C'EST UN MAITRE COUP !"  GOTO 4970
4930 IF B>6 THEN PRINT"OH, CA FAIT TRES MAL !!!"  GOTO 4970
4940 IF B>4 THEN PRINT"QUELLE FIT !"  GOTO 4970
4950 IF B>2 THEN PRINT"U. PEU FAIBLE !"  GOTO 4970
4960 PRINT"JUSTE UNE EGRATIGNURE"
4970 RETURN
5000 REM==================================================
5010 REM                 MALEDICTION !
5020 GOSUB 5200
5030 PRINT"PAR LE SANG DU DRAGON !"
5040 GOSUB 5200
5050 PRINT"PAR LE DRAGON DU CHATEAU !"
5060 GOSUB 5200
5070 PRINT QU'A L'APPEL DE SON NOM !"
5080 GOSUB 5200
5090 PRINT"LE TRESOR QUITTE VOS DS "
5100 GOSUB 5600
5110 PRINT"VOILA QUI EST FAIT, LE TRESOR A DISPARU SON COFFRE."
5120 RETURN
5200 REM==================================================
5210 REM                 MISE EN PACE
5220 GOSUB 5600
5230 PRINT"
5240 RETURN
5300 REM==================================================
5310 REM                 FIN DU JEU APRES UN COMBAT CONTRE UN MONSTRE
5320 PRINT"CE DERNIER COUP QUE VOUS A PORTE ";M$(S);" VOUS A TUE."
5330 GOSUB 5600
5340 GOSUB 5600
5350 RETURN
5400 REM==================================================
5410 REM                 CRISUT DE LA FIN
5420 IF A<>0 THEN PRINT"EXCUSEZ-MOI UN INSTANT !"  GOSUB 5000
5430 GOSUB 5900
5440 RETURN
5500 REM==================================================
5510 REM                 C'EST LA MORT !
5520 PRINT"VOTRE SQUELETTE COMME CELUI DE TANT D'AUTRES TEMOIGNERA DE"
5530 PRINT"VOTRE COURAGE, MAIS AUSSI DE VOTRE TEMERITE"
5540 GOSUB 5600
5550 PRINT"DOMMAGE !!!   VOUS M'ETIEZ SYMPATHIQUE."
5560 PRINT"MAIS PEU DE GENS ECHAPPENT DU CHATEAU DU DRAGON, HELAS !!!"
5570 RETURN
5600 REM==================================================
5610 REM                 TEMPORISATION
5620 FOR I=1 TO 500: NEXT I
5630 RETURN
5700 REM==================================================
5710 REM                 ATTENTE D'UNE COMMANDE
5720 ZX=0
5730 A$=INKEY$
5740 ZX=ZX+1
5750 IF ZX=9 THEN GOSUB 5800
5760 IF A$="" THEN 5730
5770 RETURN
5800 REM==================================================
5810 REM                 CLIGNOTEMENT DU POINT LUMINEUX
5820 ZX=0
5830 IF POINT(X,Y)=1 THEN RESET(X,Y) ELSE SET(X,Y)
5840 RETURN
5900 REM==================================================
5910 REM                 ECRITURE DES COMMENTAIRES SOUS LE PLAN DES LIEUX
5920 PRINT@32,B$(TT)
5930 QTT=1
5940 IF B$(Q)="" THEN 5960
5950 PRINT B$(Q)
5960 GOSUB 5600
5970 IF B$(Q+1)<> " THEN TT=TT+1  GOTO 5920
5980 RETURN
6000 REM==================================================
6010 REM                 MISE A JOUR DU NOMBRE DE POINTS DE VIE RESTANT
6020 PRINT@760,A1
6030 RETURN
9000 REM==================================================
9010 REM                 TEST SUR LA REPONSE
9020 INPUT A$
9030 IF A$<>"N" AND A$<>"O" THEN PRINT"REPONDRE PAR O OU N"  GOTO 9030
9040 RETURN
10000 REM=================================================
10010 DATA1,1,1,2,2,2,3,3,3,4,4,4,5,5,5,5,5,5,5,5
10020 DATA5,5,5,5,5,7,7,7,7,7,8,8,8,9,9 9,10,11,11,11,11,11,11,11,11
10030 DATA12,12,12,12,12,12,12,12,12,12 12 13,13,13 13,14,14,14,14,14,15
10040 DATA15,15,15,16,16,16,17,17,17,18,18 19,19,19,19,20,20,21,21
10050 DATA22,22,22,22,22,23,23,23,23,23,23,23,23
```

le combat	points de vie	force	résistance
vous	28	1 à 6 + 0	7
un gobelin	2	1 a 3 + 1	4

jusqu au retour de l'expedition, ou a la mort de tous ses membres

Pour aider le Maître du Donjon il existe des recueils qui donnent des idees de scenarios qui decrivent des monstres, avec leurs caracteristiques, qui répondent aux diverses situations dans lesquelles peut se trouver le Maître du Donjon Il existe enfin des figurines représentant les personnages et les monstres, ainsi que les tresors les armures etc

Dans « le château du Dragon », c'est l'ordinateur qui joue le rôle du Maître du Donjon Vous constituez l'expedition

Il s'agit partant du village d'atteindre la salle du trésor d'ouvrir le coffre et de rentrer au village vivant si possible

Vous selectionnez trois fois de suite la direction vers laquelle vous desirez aller (le terrain représente un carre de 50 x 50) l'ordinateur vous indique ou vous vous trouvez et determine éventuellement un evenement Cet évenement peut être soit benefique (objets benefiques), soit malefiques (rencontre avec un monstre) Arrive dans la salle du tresor, il faut trouver la combinaison du coffre selon le principe du Mastermind

Voici les principales caracteristiques du jeu

Les combats Chacune des créatures du « château du Dragon » (personnage ou monstre) est caracterisee par trois facteurs le capital vie la resistance et la force

Le capital-vie est la vitalite de la creature Chaque blessure le diminue, son annulation correspond à la mort

La resistance est le seuil minimal de l'ardeur que doit deployer l'adversaire pour que son coup porte La resistance d'une momie est par exemple de 8 Au cours d'un combat, vous ne toucherez la momie que si l'ordinateur réalise (par un RND(10)) un nombre supérieur ou egal a 8

La force permet de determiner la gravite de la blessure infligee après un coup reussi La force est définie par deux variables f et f La gravite du coup est calculée par RND(f) + f et est retranchee au capital-vie de l'adversaire

Grâce a ces trois principes un combat vraisemblable peut avoir lieu Une creature faible mais chanceuse peut vaincre une creature plus forte

Voici un exemple de combat

Vous êtes pres du château tout d'un coup surgit un gobelin

Le gobelin frappe le premier Il réalise un 8 8 est superieur ou egal a 7 donc le coup a porté La blessure infligee (RND(3)+1) est de 3 Votre nouveau total est donc de 28-3 = 25

A vous de frapper Vous realisez un 3 3 est inferieur a 4 donc votre coup n a pas porte Au gobelin de frapper etc jusqu a la mort de l'un des adversaires

Les principales variables sont A1 capital vie, A2 constante force A3 resistance A4 drapeau du tresor pour l'homme et M1 capital-vie M2 force M3 constante force M4 resistance pour le monstre

(x y) coordonnees du point clignotant

I J,Q indice de boucle

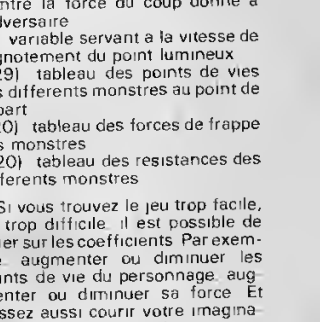
ZZ drapeau indiquant si on termine ou non le jeu

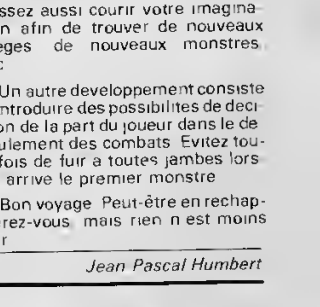
K indice de la boucle Mastermind

M$(20) tableau des monstres

G$(10) tableau des exclamations (commentaires)

I1 nombre de directions déjà demandees





FF drapeau indiquant les differentes impossibilites de se deplacer dans une direction donnee

A$ réponse pour la direction

V W variables de travail coordonnees du point lumineux

Z variable de travail niveau d'incrementation de la variable I

D,D1 D2 D3 D4,D5 drapeau indiquant si apres déplacement on se trouve dans le même lieu geographique (village lande château abords du château)

ZI drapeau pour ravitaillement au village

W(I) nombre aleatoire

E$ reponse pour proposition pour le coffre

G nombre de chiffres justes (de la combinaison du coffre)

F(I) indique si le chiffre est bien place et juste

G (I) indique si le chiffre est mal place mais existe dans la combinaison

H nombre de chiffres mal places mais existant dans la combinaison

TT initialisation de la boucle de lecture des commentaires

B$(6) chaînes de caracteres commentaires

T est egal a 9 ou 11 indiquant si on est aux abords du château ou si on est dans la lande aride

U variable aleatoire fin de la boucle de lecture des donnees en data

S donnees en data

B variable servant a la determination des nouveaux points de vie, montre la force du coup donne a l'adversaire

ZX variable servant a la vitesse de clignotement du point lumineux

N(29) tableau des points de vies des differents monstres au point de depart

Q(20) tableau des forces de frappe des monstres

Q(20) tableau des resistances des differents monstres

Si vous trouvez le jeu trop facile, ou trop difficile il est possible de jouer sur les coefficients Par exemple augmenter ou diminuer les points de vie du personnage augmenter ou diminuer sa force Et laissez aussi courir votre imagination afin de trouver de nouveaux pieges de nouveaux monstres etc

Un autre developpement consiste a introduire des possibilites de decision de la part du joueur dans le deroulement des combats Evitez toutefois de fuir a toutes jambes lorsqu arrive le premier monstre

Bon voyage Peut-être en rechapperez-vous mais rien n est moins sur

Jean Pascal Humbert

à devin, devin et demi

choisissez vos bons augures

« L'avenir n'est à personne » disait Victor Hugo. Ne laissez donc à personne le soin de le préparer pour vous : si vous avez besoin d'un petit oracle, pourquoi n'utiliseriez-vous pas un programme en BASIC pour TRS.

Il y a certaines choses qui n'ont l'air de rien mais que l'être humain n'accomplit qu'avec de grosses difficultés. On trouve même certaines activités, apparemment simples comme bonjour qui sont pratiquement impossibles à effectuer. Ainsi, par exemple (mais cet exemple n'a pas été choisi au hasard) dire *n'importe quoi*. Quels que soient les efforts que l'on consent et les précautions dont on s'entoure, ce que l'on dit a toujours un sens. Il n'est pas toujours facile de restituer ce sens intégralement mais si l'on accepte de passer un certain temps à y réfléchir, on en retrouve au moins quelques bribes.

Essayez de dire une phrase qui n'ait « ni queue ni tête », notez-la, recherchez ce qu'elle vous évoque, faites un petit effort et vous vous apercevrez sans doute qu'elle n'est pas insignifiante. Il y a en fait de fortes chances pour que cette phrase vous *parle*, et il est logique d'en conclure que, lorsqu'on dit « n'importe quoi », quelque chose nous échappe.

Certains psychologues soutiennent même l'idée selon laquelle il nous serait impossible de donner un nombre « au hasard ». En creusant un peu ce que ce nombre nous dit, on y découvrirait assez vite qu'il n'est pas si quelconque que cela. Bref, si l'on peut risquer cette formule peut-être désagréable, nous serions dotés d'un très mauvais générateur de nombres aléatoires !

Dire n'importe quoi... Essayons tout de même. En nous faisant aider par un ordinateur individuel, nous devrions presque y réussir. Sur ce type de machines en effet, on peut utiliser des séquences d'instructions qui fournissent des nombres tirés au hasard. Peu nous importe ici de savoir si les loteries arithmétiques respectent parfaitement les lois de la probabilité : il suffit que les nombres tirés ne nous soient pas connus d'avance.

Ces nombres serviront à choisir des mots, et avec ces mots le programme construira des phrases qui, très probablement, nous étonneront. Pour la plupart d'entre elles, il ne nous serait jamais venu à l'idée de les prononcer !

Comme chacun sait, l'ordinateur compte bien : nous nous arrangerons donc au passage pour lui confier la tâche de compter les syllabes des phrases et nous ferons en sorte que chacune des

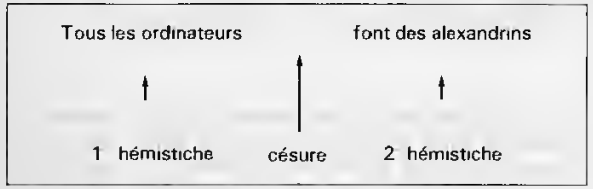
Première parution : L'OI n° 33

liste du programme sur TRS-80

```
10 ' * * * * * * * * * * * * * * *
20 ' * L'ORACLE D'ALEXANDRIE *
30 ' * * * * * * * * * * * * * * *
40 ' COPYRIGHT
50 '
60 ' L'ORDINATEUR INDIVIDUEL
70 '
80 ' * * * * * * * * * * * * * * * * * * * * * *
90 '
100 ' ---- SUBSTANTIFS MASCULINS = MS() ----
110 ' ---- SUBSTANTIFS FEMININS = FS() ----
120 ' ---- ADJECTIFS MASCULINS = GS() ----
130 ' ---- ADJECTIFS FEMININS = FS() ----
140 ' ---- VERBES AU FUTUR = VS() ----
150 ' ---- VERBES AU PRESENT = MS() ----
160 ' ---- PARTICIPES PASSES = PS() ----
170 '
180 >>>>>>>>>> DIMENSIONNEMENT DES TABLEAUX <<<<<<<<
190 '
200 DIM AS(50),MS(70),FS(50),GS(50),PS(50)
210 '
220 >>>>>>>>>>>>>>>> AFFICHAGE DU TITRE <<<<<<<<<<<<<<<<
230 '
240 FOR - 1 TO 6
250 PRINT CHR$(23)
260 FOR - 1 TO 50 : NEXT J
270 PRINT L'ORACLE D'ALEXANDRIE
280 NEXT - 1 : FOR J = 1 TO 100 : NEXT J
290 CLS
300 PRINT "BONJOUR
310 PRINT "VOUS PRIE D'ATTENDRE,
320 PRINT "VOUS PLAIT
330 PRINT DANS UN INSTANT,
340 PRINT IL VOUS SUFFIRA D'APPUYER SUR UNE TOUCHE
350 PRINT ET VOUS SAUREZ ENFIN LA VERITE
360 PRINT ENVIRONNANT, CE N'EST PAS TOUJOURS TRES CLAIR,
370 PRINT MAIS COMME ON PEUT EN GAGER AUTANT QUE L'ON VEUT,
380 PRINT >>>>>>>>>>>>>>> LECTURE DES DATA <<<<<<<<<<<<<<
390 '
400 '
410 FOR - 1 TO 50 : READ AS : NEXT J
420 REM AS = " FIN " THEN GOTO 450
430 READ MS : IF MS = " FIN " THEN GOTO 470
440 READ FS : IF FS = " FIN " THEN GOTO 490
450 READ GS : IF GS = " FIN " THEN GOTO 510
460 REM
470 READ PS : IF PS = " FIN " THEN GOTO 530
480 REM
490 READ PS : IF PS = " FIN " THEN GOTO 550
500 REM
510 REM
520 REM
530 REM
540 REM
550 REM
560 REM
570 REM
580 REM
590 REM
600 REM
610 REM
620 REM
630 REM
640 REM
650 REM
660 REM
670 REM
680 REM
690 REM
700 REM
710 REM
720 REM
730 REM
740 REM
750 REM
760 REM
770 REM
780 REM
790 REM
800 REM
810 REM
820 REM
830 REM
840 REM
850 REM
860 REM
870 REM
880 REM
890 REM
900 REM
910 REM
920 REM
930 REM
940 REM
950 REM
960 REM
970 REM
980 REM
990 REM
1000 REM
1010 REM
1020 REM
1030 REM
1040 REM
1050 REM
1060 REM
1070 REM
1080 REM
1090 REM
1100 REM
1110 REM
1120 REM
1130 REM
1140 REM
1150 REM
1160 REM
1170 REM
1180 REM
1190 REM
1200 REM
1210 REM
1220 REM
1230 REM
1240 REM
1250 REM
1260 REM
1270 REM
1280 REM
1290 REM
1300 REM
1310 REM
1320 REM
1330 REM
1340 REM
1350 REM
1360 REM
1370 REM
1380 REM
1390 REM
1400 REM
1410 REM
1420 REM
1430 REM
1440 REM
1450 REM
1460 REM
1470 REM
1480 REM
1490 REM
1500 REM
1510 REM
1520 REM
1530 REM
1540 REM
1550 REM
1560 REM
1570 REM
1580 REM
1590 REM
1600 REM
1610 REM
1620 REM
1630 REM
1640 REM
1650 REM
1660 REM
1670 REM
1680 REM
1690 REM
1700 REM
1710 REM
1720 REM
1730 REM
1740 REM
1750 REM
1760 REM
1770 REM
1780 REM
1790 REM
1800 REM
1810 REM
1820 REM
1830 REM
1840 REM
1850 REM
1860 REM
1870 REM
1880 REM
1890 REM
1900 REM
1910 REM
1920 REM
1930 REM
1940 REM
1950 REM
1960 REM
1970 REM
1980 REM
1990 REM
2000 REM
2010 REM
2020 REM
2030 REM
2040 REM
2050 REM
2060 REM
2070 REM
2080 REM
2090 REM
2100 REM
2110 REM
2120 REM
2130 REM
2140 REM
2150 REM
2160 REM
2170 REM
2180 REM
2190 REM
2200 REM
2210 REM
2220 REM
2230 REM
2240 REM
2250 REM
2260 REM
2270 REM
2280 REM
2290 REM
2300 REM
2310 REM
2320 REM
2330 REM
2340 REM
2350 REM
2360 REM
2370 REM
2380 REM
2390 REM
2400 REM
2410 REM
2420 REM
2430 REM
2440 REM
2450 REM
2460 REM
2470 REM
2480 REM
2490 REM
2500 REM
2510 REM
2520 REM
2530 REM
2540 REM
2550 REM
2560 REM
2570 REM
2580 REM
2590 REM
2600 REM
2610 REM
2620 REM
2630 REM
2640 REM
2650 REM
2660 REM
2670 REM
2680 REM
2690 REM
2700 REM
2710 REM
2720 REM
2730 REM
2740 REM
2750 REM
2760 REM
2770 REM
2780 REM
2790 REM
2800 REM
2810 REM
2820 REM
2830 REM
2840 REM
2850 REM
2860 REM
2870 REM
2880 REM
2890 REM
2900 REM
2910 REM
2920 REM
2930 REM
2940 REM
2950 REM
2960 REM
2970 REM
2980 REM
2990 REM
3000 REM
3010 REM
3020 REM
3030 REM
3040 REM
3050 REM
3060 REM
3070 REM
3080 REM
3090 REM
3100 REM
3110 REM
3120 REM
3130 REM
3140 REM
3150 REM
3160 REM
3170 REM
3180 REM
3190 REM
3200 REM
3210 REM
3220 REM
3230 REM
3240 REM
3250 REM
3260 REM
3270 REM
3280 REM
3290 REM
3300 REM
3310 REM
3320 REM
3330 REM
3340 REM
3350 REM
3360 REM
3370 REM
3380 REM
3390 REM
3400 REM
3410 REM
3420 REM
3430 REM
3440 REM
3450 REM
3460 REM
3470 REM
3480 REM
3490 REM
3500 REM
3510 REM
3520 REM
3530 REM
3540 REM
3550 REM
3560 REM
3570 REM
3580 REM
3590 REM
3600 REM
3610 REM
3620 REM
3630 REM
3640 REM
3650 REM
3660 REM
3670 REM
3680 REM
3690 REM
3700 REM
3710 REM
3720 REM
3730 REM
3740 REM
3750 REM
3760 REM
3770 REM
3780 REM
3790 REM
3800 REM
3810 REM
3820 REM
3830 REM
3840 REM
3850 REM
3860 REM
3870 REM
3880 REM
3890 REM
3900 REM
3910 REM
3920 REM
3930 REM
3940 REM
3950 REM
3960 REM
3970 REM
3980 REM
3990 REM
4000 REM
4010 REM
4020 REM
4030 REM
4040 REM
4050 REM
4060 REM
4070 REM
4080 REM
4090 REM
4100 REM
4110 REM
4120 REM
4130 REM
4140 REM
4150 REM
4160 REM
4170 REM
4180 REM
4190 REM
4200 REM
4210 REM
4220 REM
4230 REM
4240 REM
4250 REM
4260 REM
4270 REM
4280 REM
4290 REM
4300 REM
4310 REM
4320 REM
4330 REM
4340 REM
4350 REM
4360 REM
4370 REM
4380 REM
4390 REM
4400 REM
4410 REM
4420 REM
4430 REM
4440 REM
4450 REM
4460 REM
4470 REM
4480 REM
4490 REM
4500 REM
4510 REM
4520 REM
4530 REM
4540 REM
4550 REM
4560 REM
4570 REM
4580 REM
4590 REM
4600 REM
4610 REM
4620 REM
4630 REM
4640 REM
4650 REM
4660 REM
4670 REM
4680 REM
4690 REM
4700 REM
4710 REM
4720 REM
4730 REM
4740 REM
4750 REM
4760 REM
4770 REM
4780 REM
4790 REM
4800 REM
4810 REM
4820 REM
4830 REM
4840 REM
4850 REM
4860 REM
4870 REM
4880 REM
4890 REM
4900 REM
4910 REM
4920 REM
4930 REM
4940 REM
4950 REM
4960 REM
4970 REM
4980 REM
4990 REM
5000 REM
5010 REM
5020 REM
5030 REM
5040 REM
5050 REM
5060 REM
5070 REM
5080 REM
5090 REM
5100 REM
5110 REM
5120 REM
5130 REM
5140 REM
5150 REM
5160 REM
5170 REM
5180 REM
5190 REM
5200 REM
5210 REM
5220 REM
5230 REM
5240 REM
5250 REM
5260 REM
5270 REM
5280 REM
5290 REM
5300 REM
5310 REM
5320 REM
5330 REM
5340 REM
5350 REM
5360 REM
5370 REM
5380 REM
5390 REM
5400 REM
5410 REM
5420 REM
5430 REM
5440 REM
5450 REM
5460 REM
5470 REM
5480 REM
5490 REM
5500 REM
5510 REM
5520 REM
5530 REM
5540 REM
5550 REM
5560 REM
5570 REM
5580 REM
5590 REM
5600 REM
5610 REM
5620 REM
5630 REM
5640 REM
5650 REM
5660 REM
5670 REM
5680 REM
5690 REM
5700 REM
5710 REM
5720 REM
5730 REM
5740 REM
5750 REM
5760 REM
5770 REM
5780 REM
5790 REM
5800 REM
5810 REM
5820 REM
5830 REM
5840 REM
5850 REM
5860 REM
5870 REM
5880 REM
5890 REM
5900 REM
5910 REM
5920 REM
5930 REM
5940 REM
5950 REM
5960 REM
5970 REM
5980 REM
5990 REM
6000 REM
6010 REM
6020 REM
6030 REM
6040 REM
6050 REM
6060 REM
6070 REM
6080 REM
6090 REM
6100 REM
6110 REM
6120 REM
6130 REM
6140 REM
6150 REM
6160 REM
6170 REM
6180 REM
6190 REM
6200 REM
6210 REM
6220 REM
6230 REM
6240 REM
6250 REM
6260 REM
6270 REM
6280 REM
6290 REM
6300 REM
6310 REM
6320 REM
6330 REM
6340 REM
6350 REM
6360 REM
6370 REM
6380 REM
6390 REM
6400 REM
6410 REM
6420 REM
6430 REM
6440 REM
6450 REM
6460 REM
6470 REM
6480 REM
6490 REM
6500 REM
6510 REM
6520 REM
6530 REM
6540 REM
6550 REM
6560 REM
6570 REM
6580 REM
6590 REM
6600 REM
6610 REM
6620 REM
6630 REM
6640 REM
6650 REM
6660 REM
6670 REM
6680 REM
6690 REM
6700 REM
6710 REM
6720 REM
6730 REM
6740 REM
6750 REM
6760 REM
6770 REM
6780 REM
6790 REM
6800 REM
6810 REM
6820 REM
6830 REM
6840 REM
6850 REM
6860 REM
6870 REM
6880 REM
6890 REM
6900 REM
6910 REM
6920 REM
6930 REM
6940 REM
6950 REM
6960 REM
6970 REM
6980 REM
6990 REM
7000 REM
7010 REM
7020 REM
7030 REM
7040 REM
7050 REM
7060 REM
7070 REM
7080 REM
7090 REM
7100 REM
7110 REM
7120 REM
7130 REM
7140 REM
7150 REM
7160 REM
7170 REM
7180 REM
7190 REM
7200 REM
7210 REM
7220 REM
7230 REM
7240 REM
7250 REM
7260 REM
7270 REM
7280 REM
7290 REM
7300 REM
7310 REM
7320 REM
7330 REM
7340 REM
7350 REM
7360 REM
7370 REM
7380 REM
7390 REM
7400 REM
7410 REM
7420 REM
7430 REM
7440 REM
7450 REM
7460 REM
7470 REM
7480 REM
7490 REM
7500 REM
7510 REM
7520 REM
7530 REM
7540 REM
7550 REM
7560 REM
7570 REM
7580 REM
7590 REM
7600 REM
7610 REM
7620 REM
7630 REM
7640 REM
7650 REM
7660 REM
7670 REM
7680 REM
7690 REM
7700 REM
7710 REM
7720 REM
7730 REM
7740 REM
7750 REM
7760 REM
7770 REM
7780 REM
7790 REM
7800 REM
7810 REM
7820 REM
7830 REM
7840 REM
7850 REM
7860 REM
7870 REM
7880 REM
7890 REM
7900 REM
7910 REM
7920 REM
7930 REM
7940 REM
7950 REM
7960 REM
7970 REM
7980 REM
7990 REM
8000 REM
8010 REM
8020 REM
8030 REM
8040 REM
8050 REM
8060 REM
8070 REM
8080 REM
8090 REM
8100 REM
8110 REM
8120 REM
8130 REM
8140 REM
8150 REM
8160 REM
8170 REM
8180 REM
8190 REM
8200 REM
8210 REM
8220 REM
8230 REM
8240 REM
8250 REM
8260 REM
8270 REM
8280 REM
8290 REM
8300 REM
8310 REM
8320 REM
8330 REM
8340 REM
8350 REM
8360 REM
8370 REM
8380 REM
8390 REM
8400 REM
8410 REM
8420 REM
8430 REM
8440 REM
8450 REM
8460 REM
8470 REM
8480 REM
8490 REM
8500 REM
8510 REM
8520 REM
8530 REM
8540 REM
8550 REM
8560 REM
8570 REM
8580 REM
8590 REM
8600 REM
8610 REM
8620 REM
8630 REM
8640 REM
8650 REM
8660 REM
8670 REM
8680 REM
8690 REM
8700 REM
8710 REM
8720 REM
8730 REM
8740 REM
8750 REM
8760 REM
8770 REM
8780 REM
8790 REM
8800 REM
8810 REM
8820 REM
8830 REM
8840 REM
8850 REM
8860 REM
8870 REM
8880 REM
8890 REM
8900 REM
8910 REM
8920 REM
8930 REM
8940 REM
8950 REM
8960 REM
8970 REM
8980 REM
8990 REM
9000 REM
9010 REM
9020 REM
9030 REM
9040 REM
9050 REM
9060 REM
9070 REM
9080 REM
9090 REM
9100 REM
9110 REM
9120 REM
9130 REM
9140 REM
9150 REM
9160 REM
9170 REM
9180 REM
9190 REM
9200 REM
9210 REM
9220 REM
9230 REM
9240 REM
9250 REM
9260 REM
9270 REM
9280 REM
9290 REM
9300 REM
9310 REM
9320 REM
9330 REM
9340 REM
9350 REM
9360 REM
9370 REM
9380 REM
9390 REM
9400 REM
9410 REM
9420 REM
9430 REM
9440 REM
9450 REM
9460 REM
9470 REM
9480 REM
9490 REM
9500 REM
9510 REM
9520 REM
9530 REM
9540 REM
9550 REM
9560 REM
9570 REM
9580 REM
9590 REM
9600 REM
9610 REM
9620 REM
9630 REM
9640 REM
9650 REM
9660 REM
9670 REM
9680 REM
9690 REM
9700 REM
9710 REM
9720 REM
9730 REM
9740 REM
9750 REM
9760 REM
9770 REM
9780 REM
9790 REM
9800 REM
9810 REM
9820 REM
9830 REM
9840 REM
9850 REM
9860 REM
9870 REM
9880 REM
9890 REM
9900 REM
9910 REM
9920 REM
9930 REM
9940 REM
9950 REM
9960 REM
9970 REM
9980 REM
9990 REM
10000 REM
10010 REM
10020 REM
10030 REM
10040 REM
10050 REM
10060 REM
10070 REM
10080 REM
10090 REM
10100 REM
10110 REM
10120 REM
10130 REM
10140 REM
10150 REM
10160 REM
10170 REM
10180 REM
10190 REM
10200 REM
10210 REM
10220 REM
10230 REM
10240 REM
10250 REM
10260 REM
10270 REM
10280 REM
10290 REM
10300 REM
10310 REM
10320 REM
10330 REM
10340 REM
10350 REM
10360 REM
10370 REM
10380 REM
10390 REM
10400 REM
10410 REM
10420 REM
10430 REM
10440 REM
10450 REM
10460 REM
10470 REM
10480 REM
10490 REM
10500 REM
10510 REM
10520 REM
10530 REM
10540 REM
10550 REM
10560 REM
10570 REM
10580 REM
10590 REM
10600 REM
10610 REM
10620 REM
10630 REM
10640 REM
10650 REM
10660 REM
10670 REM
10680 REM
10690 REM
10700 REM
10710 REM
10720 REM
10730 REM
10740 REM
10750 REM
10760 REM
10770 REM
10780 REM
10790 REM
10800 REM
10810 REM
10820 REM
10830 REM
10840 REM
10850 REM
10860 REM
10870 REM
```

phrases aient douze syllabes. Pour être tout à fait précis, ce n'est pas la syllabe que nous ferons décompter, mais — alexandrin oblige — c'est le pied.

Dans cette version du programme, nous nous contenterons d'un alexandrin très classique, régulièrement coupé en deux parties égales (en versification, on parle d'hémistiches) séparées par une petite pause vocale (le césure). Essayez de vous souvenir, je suis certain que vous avez appris tout cela durant votre scolarité. Si vous avez complètement oublié, ne vous faites pas de souci ; voici un exemple d'alexandrin classique et régulier (mais pas très fameux).



Ce type de vers, qui est un des plus répandus de la poésie française, a son rythme bien à lui, il s'en dégage une petite musique, une *scansion* particulière, tarata-ratata (silence) tarataratata, soit six pieds plus six pieds, ce qui fait douze. Le compte est bon. Naturellement, il y a des variantes sur le même schéma rythmique :

tatarata-tata/taratata-tata
tatara-tatara/tata-taratata
tata-taratata/tarata-tarata
etc.

et je vous fais grâce du reste car le programme s'en chargera très bien.

exemple

LE MOULIN DU REVEIL
UN LUTIN DE MISERE IRA VERS LA DOUCEUR.

SIRENE DE L'IRONIE
LE REVEUR IMPORTUN A SURPRIS LA BEAUTE.

LA MINUTE DE L'AFFECTION
UN MALHEUR TRANSPARENT IMPLORE LA MALICE.

REGARD DU DEDAIN
UN DESIR TOUT A COUP A CASSE L'AMITIE.

DEESSE DE LA RIVIERE
UN CONTEUR RAYONNANT ACCUEILLE SA CHANSON.

Construire des alexandrins par milliers

Pour ne pas repousser trop loin les limites du raisonnable, j'ai décidé que le vocabulaire utilisé serait composé de quelques centaines de mots et que tous les vers fournis par l'ordinateur seraient « coulés » dans le même moule syntaxique : sujet / verbe / complément.

Que le vocabulaire ne soit pas très étendu n'a guère d'importance, à condition bien entendu, qu'il soit bien choisi. Vous pouvez d'ailleurs le modifier au gré de votre fantaisie et le faire grossir autant que vous le permettra la place disponible dans la « mémoire » de votre machine.

Le seconde restriction est beaucoup plus fâcheuse, car elle conduit en fait à donner à tous ces alexandrins automatiques un air de famille. Chacune des phrases étant bâtie sur le même modèle grammatical, il est difficile d'en aligner quatre ou cinq d'affilée sans engendrer une impression passablement désagréable de monotonie. Il n'était donc pas question de s'essayer à produire des épopées. Et je n'aborde pas le problème de la rime !

Sachant par ailleurs que le but du programme était de produire des textes qui signifient *n'importe quoi*, j'ai pensé qu'il fallait mieux ne pas s'étendre. C'est bien connu, ce sont les plaisanteries les meilleures. Je me suis donc modestement contenté de poèmes qui ne comptent qu'un seul vers. Cela se fait en Chine et au Japon : quelques mots qui vous font on ne sait trop quoi, vous les écoutez et cela vous laisse tout chose.

Seulement voilà, dans le programme en question, ce n'est ni comme en Chine ni comme au Japon. Ici, tout se fait au hasard, et même si parfois le hasard fait bien les choses, ce n'est pas la règle générale. Certains alexandrins sont d'une banalité affligeante (j'aurais pu les écrire moi-même, c'est dire), d'autres sont assez jolis : on y comprend ce que l'on veut et c'est très bien. D'autres encore, les plus nombreux, sont obscurs.

Avec beaucoup de patience, on parvient à « tirer un vers » excellent de loin en loin. L'ordinateur est infatigable, mais pour mériter ce petit joyau de douze pieds qui survient on ne sait pourquoi, on doit s'infliger la lecture d'une longue série d'alexandrins très hermétiques. Pour l'avoir pratiquée plusieurs heures, je sais que ce jeu est assez fastidieux à la longue. Et c'est la raison pour laquelle je vous en propose un autre. Au demeurant, cela ne vous privera pas des rares alexandrins bien trouvés qui peuvent toujours survenir.

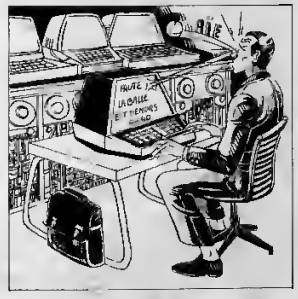
Puisqu'elles sont les plus nombreuses, intéressons-nous aux phrases obscures. Ces vers sibyllins ont en effet toutes les caractéristiques des oracles : le sens en est obscur et ils disent n'importe quoi. Chacun peut y lire son avenir ou celui des autres, chacun peut y lire ce qu'il veut. Vous voici devin. La Pythie de Delphes, qui s'était fait une solide réputation en la matière, ne procédait pas autrement. Il suffit de dire n'importe quoi, et si la phrase impénétrable est parée des vertus de l'oracle, il se trouvera toujours quelqu'un pour lui donner un sens, pour l'orienter, il se trouvera toujours une situation à venir où elle deviendra limpide.

N'allez pas toutefois en faire le commerce : la loi française, à juste titre, a prévu des peines pour quiconque abuserait de la crédulité publique. Et je crois qu'elle a très bien fait.

Vostradamus

le tennis dans un fauteuil

Il y a beaucoup de jeux pour ordinateurs individuels qui occupent plusieurs personnes, preuve que l'ordinateur n'isole pas nécessairement, mais qu'il peut bien au contraire réunir. Lorsque votre écran se transforme en un court de tennis, vous ne jouez pas avec ou contre la machine, vous jouez grâce à elle, et entre vous : le tennis est par excellence un jeu où l'on se renvoie la balle. Pour utiliser ce programme de tennis, vous n'aurez besoin ni de balles ni de raquettes, mais vous pouvez prévoir deux sièges confortables.



On retrouvera ici les principes sur lesquels reposent certains jeux « vidéo » : deux raquettes se déplacent verticalement aux deux extrémités de l'écran. Dans l'entre-deux une balle vient rebondir de l'une à l'autre et sur les côtés supérieur et inférieur de l'écran jusqu'à ce que l'un des joueurs la laisse passer derrière la raquette, ce qui lui fait bien entendu perdre un point.

Que penserait Borg des balles à vitesse réglable ?

Le programme a été conçu sur un TRS 80 niveau 2 et il occupe environ 3 Ko en mémoire.

Pour les mouvements de la raquette vers le haut et vers le bas, chacun des deux adversaires dispose de deux touches. Celui de gauche appuie sur les touches portant les flèches verticales (vers le haut et vers le bas : sans commentaire). Celui de droite obtient le même résultat en enfoncant respectivement les touches numériques 9 (vers le haut) et 6 (vers le bas). Vous constaterez qu'on s'y habitue très vite.

Pour que tout le monde puisse trouver son bonheur, le joueur averti comme celui qui n'a jamais touché une raquette, on a prévu pour la balle une vitesse réglable.

Les deux adversaires choisissent cette vitesse au début de la partie (ligne 300 à 350) en indiquant un chiffre compris entre 2 et 5. Cette valeur est rangée dans une variable C et représente l'intervalle, en abscisse, de l'affichage de la balle. Avec 2, les échanges sont très calmes. Avec 5, on s'envoie des « boulets de canon ».

Le graphisme du TRS 80 modèle I (une grille de 128 x 48) ne permet pas d'accroître l'intervalle des déplacements sur l'axe des ordonnées : il s'effectuera toujours avec un pas égal à l'unité. Une valeur supérieure transformerait notre tennis en une sorte de billard.

C'est aux lignes 600 à 1600

Première parution : L'OI n° 30

```
10 REM                    * TENNIS *
20 REM              MIS AU POINT SUR TRS-80
30 REM  AUTEUR : MARC AUBRY
40 REM  COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
50 REM*****************************************************************
100 CLS
110 GOSUB 1000                  : REM PRESENTATION
120 GOSUB 1200                    REM MODE D'EMPLOI
130 GOSUB 1400                    REM CHOIX DES VITESSES
140 GOSUB 1600                  : REM AFFICHAGE DU JEU
150 GOSUB 2700                    REM AFFICHAGE DES RAQUETTES
160 IF M1=10 OR M2=10 THEN 2A0
170 A1=25: A2=25: N1=0: N2=0
180 IF J=1 THEN X=H2: GOTO 220
190 X=H1
200 IF U=2 THEN GOSUB 1700 ELSE GOSUB 1900    : REM DEPLACEMENT DE LA BALLE
210 GOTO 150
220 IF U=2 THEN GOSUB 2000 ELSE GOSUB 2200    : REM DEPLACEMENT DE LA BALLE
230 GOTO 150
240 PRINT: PRINT"VOULEZ-VOUS UNE AUTRE PARTIE ?"
250 GOSUB 9000                    REM TEST SUR LA REPONSE
260 IF RS="O" THEN 130
270 END
1000 REM*****************************************************************
1010 REM                  PRESENTATION DU JEU
1020 FOR B=1 TO 127
1030     SET(B,18)
1040 NEXT B
1050 PRINTBA48,CHR$(23)"* TENNIS *"
1060 FOR B=1 TO 127
1070     SET(B,25)
1080 NEXT B
1090 FOR B=1 TO 3500. NEXT B
1100 RETURN
1200 REM-----------------------------------------------------------------
1210 REM                  MODE D'EMPLOI
1220 CLS
1230 PRINT CHR$(23),
1240 PRINTB130,"POUR DEPLACER LA RAQUETTE,"
1250 PRINTB258,"LE JOUEUR DE GAUCHE DOIT"
1260 PRINT"APPUYER SUR LA FLECHE EN HAUT"
1270 PRINT'      OU SUR LA FLECHE EN BAS."
1280 FOR B=1 TO 1500: NEXT B
1290 PRINTB514,"LE JOUEUR DE DROITE DOIT"
1300 PRINT' APPUYER SUR 9 POUR LE HAUT
1310 PRINT'       OU POUR 6 POUR LE BAS
1320 FOR B=1 TO 3500. NEXT B
1330 RETURN
1400 REM-----------------------------------------------------------------
1410 REM                  CHOIX DES VITESSES
1420 CLS
1430 M1=0: M2=0
1440 PRINT CHR$(23),
1450 PRINTB204,"CHOISISSEZ VOTRE VITESSE"
1460 PRINTB404,"LENTE .............2"
1470 PRINTB668,"MOYENNE............3"
1480 PRINTB532,"RAPIDE ............4"
1490 PRINTB594,"SUPERSONIQUE ......5"
1500 PRINTB782,"VOTRE CHOIX";: INPUT C
1510 IF C>5 OR C<2 THEN 1400
1520 IF C=2 THEN M1=10: M2=113: B=122: F=6
1530 IF C=3 THEN M1=10: M2=118: B=121: F=7
1540 IF C=4 THEN M1=12. M2=116: B=120: F=8
1550 IF C=5 THEN M1=9: M2=119: B=119: F=9
1560 RETURN
1600 REM-----------------------------------------------------------------
1610 REM                  AFFICHAGE DU JEU
1620 CLS
1630 E1=4: E2=124
1640 FOR B=1 TO 127
1650     SET(B,0)
1660     SET(B,47)
1670 NEXT B
1680 RETURN
1700 REM-----------------------------------------------------------------
1710 REM                  DEPLACEMENT DE LA BALLE
1720 REM          En bas a droite
1730 X=X+C: Y=Y+1
1740 SET(X,Y): RESET(X-C,Y-1)
1750 IF X=E2 THEN M2=M2+1: GOTO 2320
1760 GOSUB 2400
1770 IF NOT POINT(X+C,Y+1) THEN 1730
1780 IF POINT(X+C,Y) THEN 1800
1790 IF POINT(X,Y+1) THEN 1900 ELSE 2000
1800 IF X=B AND Y=46 THEN 2000
1810 IF X=B THEN 2200
1900 REM-------- en haut a droite ------------------------------------
1910 X=X+C: Y=Y-1
1920 SET(X,Y): RESET(X-C,Y+1)
1930 IF X=E2 THEN M2=M2+1: GOTO 2320
1940 GOSUB 2400
1950 IF NOT POINT(X+C,Y-1) THEN 1910
1960 IF POINT(X+C,Y) THEN 1990
1970 IF POINT(X,Y-1) THEN 1700 ELSE 2200
1980 IF X=B AND Y=1 THEN 2200
1990 IF Y=1 THEN 1700
2000 REM------- en haut a gauche ------------------------------------
2010 X=X-C: Y=Y-1
2020 SET(X,Y): RESET(X+C,Y+1)
2030 IF X=E1 THEN M1=M1+1: J=1: GOTO 2320
2040 J=0
2050 GOSUB 2400
2060 IF NOT POINT(X-C,Y-1) THEN 2010
2070 IF POINT(X-C,Y) THEN 2050
2080 IF POINT(X,Y-1) THEN 2200 ELSE 1700
2090 IF X=F AND Y=1 THEN 1700
2100 IF X=F THEN 1900
2200 REM------ en bas a gauche ------------------------------------
2210 X=X-C: Y=Y+1
2220 SET(X,Y): RESET(X+C,Y-1)
2230 IF X=E1 THEN M1=M1+1: J=1: GOTO 2320
2240 J=0
2250 GOSUB 2400
2260 IF NOT POINT(X-C,Y+1) THEN 2310
2270 IF POINT(X-C,Y) THEN 2290
2280 IF POINT(X,Y+1) THEN 2000 ELSE 1900
2290 IF X=F AND Y=46 THEN 1900
2300 IF Y=46 THEN 2000
2310 GOTO 1700
2320 RETURN
2400 REM-----------------------------------------------------------------
2410 REM                  DEPLACEMENT DE LA RAQUETTE GAUCHE
2420 IF N1=0 AND PEEK(14409)=8 THEN A=23
2430 N1=N1+1
2440 IF PEEK(14409)<>16 THEN 2460
2450 A1=A1+1
2460 IF A1=47 THEN A1=A1-1: GOTO 2480
2470 SET(4,A1): RESET(4,A1-3)
2480 IF PEEK(14409)<>8 THEN 2520
2490 A1=A1-1
2500 IF A1=2 THEN A1=A1+1: GOTO 2520
2510 SET(4,A1+2): RESET(4,A1+1)
2520 REM                  DEPLACEMENT DE LA RAQUETTE DROITE
2530 IF N2=0 AND PEEK(14592)=2 THEN A=23
2540 N2=N2+1
2550 IF PEEK(14357)<>64 THEN 2590
2560 A2=A2+1
2570 IF A2=47 THEN A2=A2-1: GOTO 2590
2580 SET(124,A2): RESET(124,A2-3)
2590 IF PEEK(14392)<>2 THEN 2630
2600 A2=A2-1
2610 IF A2=2 THEN A2=A2+1: GOTO 2630
2620 SET(124,A2-2): RESET(124,A2+1)
2630 RETURN
2700 REM-----------------------------------------------------------------
2710 REM                  AFFICHAGE DES RAQUETTES ET DU SCORE
2720 FOR B=1 TO 46
2730     RESET(4,B)
2740     RESET(124,B)
2750 NEXT B
2760 U=RND(2): Y=22
2770 FOR A1=21 TO 23
2780     SET(4,A1)
2790 NEXT A1
2800 FOR A2=21 TO 23
2810     SET(124,A2)
2820 NEXT A2
2830 PRINTB88,M2,"PRINTB100,M1;
2840 RETURN
9000 REM-----------------------------------------------------------------
9010 REM                  TEST SUR LA REPONSE
9020 RS=INKEY$
9030 IF RS="" THEN 9020
9040 IF RS<> "O" AND RS<>"N" THEN 9020
9050 RETURN
```

qu'est déterminée la direction de la balle. Le court de tennis occupe toute la hauteur de l'écran (ordonnées 0 à 47) et presque toute sa largeur (4 à 124). Lorsque la balle heurte les bordures inférieure ou supérieure du terrain, ou encore une raquette, elle rebondit avec un angle variant entre 90 et 140° en fonction de sa vitesse. Aux lignes 750, 950, 1 150 et 1 350, l'instruction POINT vérifie si le prochain segment situé sur le parcours de la balle est ou non allumé. Si c'est le cas, le rebond s'effectue, sinon la balle continue sur sa trajectoire.

Pour ajouter un peu d'imprévu dans le déroulement des parties, on a réservé un cas où la balle rebondit de tout autre façon : lorsqu'elle heurte l'extrémité d'une raquette, elle repart bien en sens inverse, mais selon la même droite (dans le jargon des joueurs de tennis, vous avez fait un « bois »).

Ce faux-rebond est soigneusement organisé aux lignes 770, 970, 1 170 et 1 370.

Lorsque la balle franchit les limites extrêmes du jeu sans rencontrer de segments allumés (ni raquette, ni bordure), le programme saute à la ligne 2 000 pour le départ d'une nouvelle balle. Naturellement, l'adversaire marque un point. Les raquettes reviennent à leur position de départ et le programme lance la balle dans une direction qui est laissée à la fantaisie de l'instruction RND (ligne 2 030), autant dire que l'engagement se fait n'importe comment et qu'il faut être sur ses gardes !

Tableau des variables

X, Y	Coordonnées de la balle
M, M,	Scores des deux joueurs
A, A	Déplacements des deux raquettes
H, H,	Coordonnées de départ de la balle
D, F	Limites de renvoi de la balle
U	Direction de départ de la balle
8	Affichage des limites inférieure et supérieure du jeu
C	Vitesse de la balle
E, E,	Abscisses des sorties de balles

Le déplacement des raquettes est assuré par les sous-programmes débutant aux lignes 1 500 et 1 800. Grâce à l'instruction PEEK qui scrute l'état des touches réservées à chaque joueur, la raquette se déplace aussi longtemps que la touche reste enfoncée.

Deux conseils pour terminer.

Tout d'abord, les personnes qui n'ont aucune expérience de ce type de jeux pourront, bien entendu, modifier le programme de façon à se munir d'une raquette immense, ils diminueront d'autant le risque de rater la balle.

Enfin, dans sa version actuelle, le programme se contente de compter les points un par un. A gagné le premier joueur qui totalise 10. Mais au tennis, comme chacun sait, on compte d'abord de 15 en 15, puis ça se complique, et je ne parle pas des sets et des manches. Rien ne vous empêche de faire en sorte que l'affichage des résultats respecte les règles en vigueur à Roland-Garros. A votre service.

Marc Aubry

1	ORIC	PAL 64K	2480F 2190 TTC	7	SEIKO	GP 250 X	net 3450 TTC
2	SHARP	MZ 720 64K	3500F 2995 TTC	8	SEIKO	GP 700 graphique couleur	5000 TTC
3	SHARP	MZ 80A 32K	7600F 5400 TTC	9	BECY	interfaces, logiciels CPM, maintenance	
4	SHARP	MZ 80B 32K	12600F 7500 TTC	10	BIRDY CENTRONICS EPSON NEC TRL WATANABE		
5	SHARP	PC1251 de poche	net 1490 TTC				
6	SHARP	MZ 3541 128K, Floppy	26000 HT				

* Garantie un an. Port 100 F jusqu'à 2500 F, 180 F au delà

PROFITEZ de la guerre des constructeurs !! du 1er au 31 décembre 83

SHARP BECY INFORMATIQUE
CONSEIL-SERVICE
Location-Leasing
78, Bd Montebello - 59000 LILLE - Tél. (20) 92 33 06
ouvert du lundi au vendredi de 9 à 12 h et de 14 à 18 heures

COUPON RÉPONSE à retourner à BECY accompagné du règlement
1 2 3 4 5 6 7 8 9 10 DOC (cocher la case correspondante)
Nom Adresse Signature
Code Postal Ville

Pour toutes précisions sur la société ou le produit présenté ci-dessus : Référence 109 du service-lecteurs (page 172)

les minotaures du labyrinthe

Ce mois-ci, nous vous proposons de revivre l'une des plus palpitantes aventures de Thésée. Vous vous souvenez, Thésée : celui qui avait réussi à se faire un ami d'Hercule. Rien ne l'arrêtait, celui-là. Un jour, il se met dans la tête de débarrasser la Crète d'un horrible monstre – il n'y a pas d'autre mot – qui dévorait une douzaine de jeunes athénien(ne)s chaque année, et voilà notre Thésée parti pour la Crète. Cette extraordinaire histoire, vieille de plusieurs milliers d'années, nous a été transmise par le bouche à oreille, et il en existe plusieurs versions. Celle dont est inspiré le programme pour PET proposé ici n'est pas la plus connue.

En ce temps-là, il n'y avait pas un Minotaure, mais des Minotaures. Beaucoup de Minotaures. Et ce n'est pas étonnant quand on sait que ces créatures mi-homme, mi-taureau, étaient aussi bêtes que fortes : il en mourait beaucoup. En fait, il s'agissait de brutes défonçant les murs du labyrinthe, tellement gloutonnes qu'elles ne pouvaient pas se retenir de dévorer tout ce qu'elles trouvaient. En s'y prenant bien, on avait quelques chances d'en empoisonner.

En réalité, quand Thésée est allé en découdre dans le labyrinthe, Ariane n'était pas au bout du fil et pour cause : sa mère, Pasiphaé (qui était une spécialiste de la broderie) n'aimait pas du tout qu'on touche à ses affaires et elle enfermait toutes ses bobines sous clé avec un soin qui confinait à la manie.

Il faut savoir aussi que dans ce curieux palais, les placards des communs regorgeaient de boulettes de nourriture pour animaux domestiques (il fallait bien alimenter les minotaures). Seulement, le roi Minos, le mari de la brodeuse, était mauvais payeur : il ne réglait jamais ses dettes. Ses fournisseurs excédés à la longue, lui réservaient toute leur marchandise avariée. Certaines boulettes étaient tellement gâtées qu'elles étaient franchement toxiques. C'est donc les bras chargés de ces mauvaises boulettes que Thésée pénétra dans le labyrinthe. Ariane lui en jetait de nouvelles toutes les minutes par-dessus les murs du labyrinthe et Thésée n'avait plus qu'à les laisser bien en vue derrière lui, et dès qu'un minotaure en trouvait une, il l'engloutissait et ne s'en relevait pas.

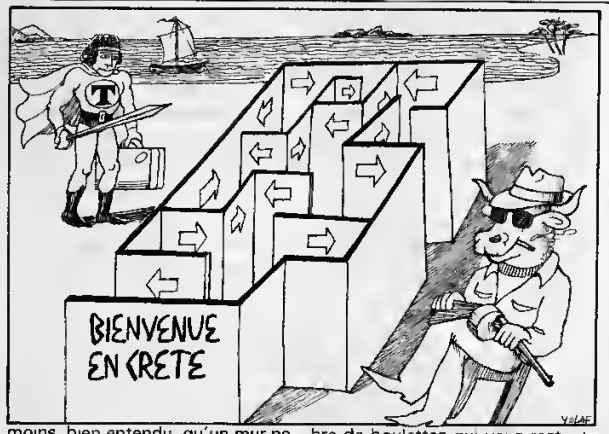
C'est exactement ce qui se passe dans notre programme. Au début du jeu, nous voyons se dessiner sous nos yeux un labyrinthe dont les détours et les recoins sont choisis au hasard (sous-programme 1300-1870). Apparaissent ensuite deux boulettes de nourriture empoisonnée (symbole +), Thésée lui-même (◆) et le premier minotaure (*). Puis une petite musique nous annonce que la poursuite infernale débute dans le dédale. Bien entendu, pour ceux qui le désirent, le programme est entièrement sonorisé. Le monstre ne reculera devant rien pour essayer de rattraper Thésée. Il n'a qu'une idée en tête : dévorer de la chair fraîche, et la faim le travaille tellement qu'il lui arrive d'emprunter des raccourcis fracassants, à certains moments, il fonce tête baissée dans le mur et passe à travers les gravats.

Naturellement, le plus excitant de l'histoire, c'est que vous êtes à la place de Thésée. Pour vous déplacer, c'est à-dire pour fuir, vous disposez des touches numériques 1 à 9.

Le 5 vous immobilise. Quand vous vous déplacez dans une direction donnée (les diagonales sont possibles), il est inutile de maintenir enfoncée la touche correspondante : vous continuerez à vous déplacer tant que vous n'appuierez pas sur une autre touche.

```
10 REM     * LE VOYAGE EN CRETE *
20 REM AUTEUR   LIONEL ANCELET
30 REM COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
40 REM*****************************************************************
100 REM LES "REM" INDIQUENT LES INSTRUCTIONS
110 REM A INTRODUIRE DANS LE PROGRAMME POUR
120 REM OBTENIR LA PARTIE SONORE DU JEU.
130 REM ILS NE SONT PAS NECESSAIRES
140 REM
200 REM*********************************
210 GOSUB 1000   REM PRESENTATION
220 POKE 1,0  POKE 2,0
230 O=PEEK(50003)
240 POKE 59490,40
250 PRINT"VISIBLE (O/N) ?"
260 GOSUB 9000  REM TEST SUR LA REPONSE
270 IF RS= "O" THEN Z=0  GOTO 290
280 Z=1
290 R=51 N=0: Q=0
300 N=90  M=42  B=43
310 GOSUB 1300  REM DESSIN DU LABYRINTHE
320 F=3*Z*17
330 REM=========
340 FOR I=1 TO Z+1*6
350 GOSUB 1900  REM BOULETTE
360 NEXT I
370 REM=========
380 GOSUB 2000  REM POSITION INITIALE DE L'HOMME
390 GOSUB 2100  REM POSITION INITIALE DU MONSTRE
400 NS=PEEK(1)*256+PEEK(2)
410 PRINT"TEMPS-          MONSTRES DETRUITS-"
420 L=24: C=0
430 GOSUB 2200  REM FORCAGE DU CURSEUR
440 PRINT" BOITES DE POISON        SCORE: "
450 IF Z=0 THEN 510
460 REM=========
470 FOR A=32768 TO 33767
480 IF PEEK(A)=102 THEN POKE A,96
490 NEXT A
500 REM=========
510 T1=2000: GOSUB 8300
520 GOSUB 2300  REM MUSIQUE INDIQUANT LE DEBUT DU JEU
530 T1=" 000000"
540 GOSUB 2400  REM AFFICHAGE DU SCORE
550 GOSUB 2600  REM DEPLACEMENT DE L'HOMME
560 GOSUB 3000  REM DEPLACEMENT DU MONSTRE
570 IF FF=0 THEN 550
580 RUN 230
590 END
1000 REM*****************************************************************
1010 REM          PRESENTATION DU JEU
1020 PRINT"
1030 PRINT:PRINT:PRINT
1040 PRINT"                MOUVEMENTS
1050 PRINT"
1060 PRINT"
1070 PRINT"                7   8   9
1080 PRINT"
1090 PRINT"                 \ | /
1100 PRINT"                  \ | /
1110 PRINT"
1120 PRINT"                4 -- 5 -- 6
1130 PRINT"
1140 PRINT"                 / | \
1150 PRINT"                / | \
1160 PRINT"
1170 PRINT"                1   2   3
1180 PRINT:PRINT:PRINT
1190 PRINT"POUR LACHER UNE BOULETTE TAPEZ SUR
1200 PRINT"          LA BARRE D'ESPACEMENT"
1210 PRINT:PRINT
1220 PRINT"APPUYER SUR RETURN POUR COMMENCER"
1230 PRINT" LE JEU"
1240 GET RS
1250 IF RS="" THEN 1240
1260 R=ASC(RS)
1270 IF R<>13 THEN 1240
1280 POKE 158,0
1290 RETURN
1300 REM=========================================================
1310 REM          LABYRINTHE
1320 AS= "&&&
```

```
3000 REM-------------------------------------------------------------
3010 REM          DEPLACEMENT DU MONSTRE
3020 FF=0
3030 U=U-SGN(X-U)
3040 V=SGN(Y-V)
3050 G=0
3060 UP=U*DU: VP=V*DV
3070 K=PEEK(32768+UP+40*VP)
3080 IF K=V* THEN POKE 32768+UP+40*VP,102
3090 IF K=M THEN GOSUB 4000: FF=1: GOTO 3300
3100 IF K=B THEN GOSUB 3700: GOTO 3300
3110 IF K=32 THEN 3210
3120 IF K<>102 AND K<>96 THEN 3150
3130 IF PEEK(32768+U+40*V)=32 THEN UP=U: GOTO 3210
3140 IF PEEK(32768+UP+40*V)=32 THEN VP=V: GOTO 3210
3150 G=1: REM REPL.ALEATOIRE
3160 DU=1: DV=0
3170 IF RND(TI)<.5 THEN DU=0: DV=1
3180 IF RND(TI)<.5 THEN DU=-DU: DV=-DV
3190 IF RND(TI)>.1 THEN 3080
3200 UP=U+DU: VP=V+DV
3210 U=UP: V=VP
3220 BP=32768+U+40*V
3230 POKE B,32
3240 B=BP
3250 POKE B,M
3260 REM*POKE 59467,16: POKE 59466,15: FOR I=200 TO 210
3270 REM*POKE 59466,I: POKE 59464,255-I: NEXT I: POKE 59467,0
3280 IF G=1 AND RND(TI)<.1 THEN 3160
3290 GOSUB 2600
3300 RETURN
3400 REM-------------------------------------------------------------
3410 REM          MUSIQUE INDIQUANT UNE BOULETTE LACHEE
3420 REM*POKE 59467,16: POKE 59466,250
3430 REM-------
3440 FOR I=1 TO 10
3450 IF PEEK(59464)=0 THEN 3470: REM*POKE 59466,15 AVANT SAUT EN 3470
3460 REM*POKE 59466,0
3470 NEXT I
3480 REM-------
3490 REM*POKE 59467,0
3500 RETURN
3600 REM-------------------------------------------------------------
3610 REM          MONSTRE MANGE UNE BOULETTE
3620 REM*POKE 59467,16: POKE 59466,15
3630 REM*POKE 59466,80: FOR I=1 TO 100: NEXT I
3640 REM*POKE 59466,250: FOR I=1 TO 100: NEXT I
3650 REM*POKE 59466,90: FOR I=1 TO 100: NEXT I
3660 REM*POKE 59467,0: FOR I=1 TO 100: NEXT I
3670 RETURN
3700 REM-------------------------------------------------------------
3710 REM          MONSTRE MANGE UNE BOULETTE
3720 POKE B,32
3730 B=32768+UP+40*V
3740 TS=TI: N=N+1
3750 GOSUB 2600
3760 REM*POKE 59467,16: POKE 59466,15
3770 REM-------
3780 FOR I=1 TO 5
3790 POKE B,40
3800 REM*FOR I=250 TO 200 STEP-1: POKE 59464,I: NEXT I
3810 POKE B,61
3820 REM*FOR I=200 TO 250: POKE 59464,I: NEXT I
3830 NEXT J
3840 REM-------
3850 POKE B,46
3860 REM*FOR I=200 TO 1 STEP-1: POKE 59466,I: NEXT I
3870 POKE B,32
3880 REM*POKE 59467,0
3890 GOTO 3100
3900 IF T>2000: GOSUB 4300
3910 TS=TS+T
3920 GOSUB 3600
3930 RETURN
4000 REM-------------------------------------------------------------
4010 REM          MONSTRE MANGE L'HOMME
4020 POKE B,32
4030 B=32768+UP+40*V
4040 REM*POKE 59467,16: POKE 59466,15
4050 REM-------
4060 FOR J=1 TO 50
4070 IF PEEK(B)=127 THEN POKE B,255: GOTO 4110
4080 REM*POKE 59466,100 AVANT LE SAUT EN 4110
4090 POKE B,127: REM*POKE 59464,200
4100 I=20: GOSUB 4300
4110 NEXT J
4120 REM-------
4130 POKE B,87
4140 REM*POKE 59467,0
4150 I=1000: GOSUB 4300
4160 GOSUB 2600
4170 IF S>HS THEN HS=S
4180 POKE 2,INT(HS/256)
4190 POKE 1,HS-256*PEEK(2)
4200 PRINT"FIN DE PARTIE"
4210 PRINT"***TOTAL"HS"***"
4220 I=2000: GOSUB 4300
4230 RETURN
4300 REM-------------------------------------------------------------
4310 REM          TEMPORISATION
4320 FOR I=1 TO I
4330 NEXT I
4340 RETURN
9000 REM-------------------------------------------------------------
9010 REM          TEST SUR LA REPONSE
9020 GET R$
9030 IF R$="" THEN 9020
9040 IF R$<>"O" AND R$<>"N" THEN 9020
9050 RETURN
```



moins, bien entendu, qu'un mur ne vous arrête. A la différence des minotaures, vous n'êtes pas un passe-muraille. Quand vous avancez, il vous suffit d'appuyer sur SPACE pour laisser tomber une boulette d'aliment toxique. Mais ne faites qu'une chose à la fois, une seule touche à la fois, et allez votre chemin de petit poucet en semant vos boulettes.

Sur l'écran, un compteur vous indiquera en permanence le nombre de boulettes qui vous restent. Bien entendu, vous pouvez ramassez une boulette qui traîne par terre : il vous suffit de passer dessus.

Le monstre, lui, réagit tout autrement : dès qu'il passe sur une boulette, il la dévore. Vous le voyez se tordre de douleur et vous l'entendez gémir. Puis il disparaît, mais il est aussitôt remplacé par un autre monstre qui surgit à un endroit imprévu du labyrinthe (par une porte dérobée). Le jeu continue.

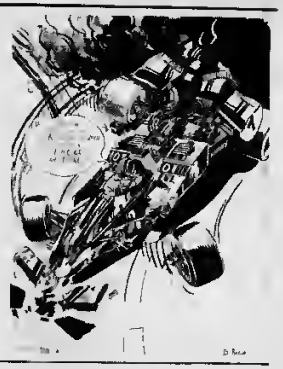
Votre score est en permanence affiché en bas et à droite de l'écran ; il est égal au nombre de monstres que vous avez empoisonnés multiplié par le nombre de secondes pendant lesquelles vous avez survécu !

Si vous recherchez vraiment la difficulté, le programme vous offre une option « à l'aveuglette » : vous pouvez, si vous le désirez, avoir un labyrinthe invisible. Au tout début de la partie, le labyrinthe se dessinera sous vos yeux, puis s'effacera comme par enchantement (sic !). A partir de ce moment, chaque fois que vous ou le monstre, butterez contre un mur, le morceau de mur apparaîtra. Dans cette version, le jeu est beaucoup plus difficile, vous aurez donc plus de boulettes.

Il ne nous reste plus qu'à vous souhaiter une bonne chance dans votre chasse aux monstres. Ne vous affolez surtout pas : on est beaucoup plus efficace quand on conserve son sang-froid, même s'il s'agit de fuir devant un monstre affamé qui vous trouve très appétissant.

Lionel Ancelet

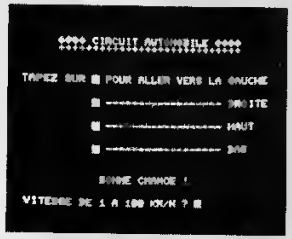
les joies de la piste (et les feux de la rampe !)



Prenez un crayon magique, dessinez un circuit automobile sur votre écran Pet/CBM. Un labyrinthe de préférence plus un autre encore plus compliqué et enfin, un parcours suicidaire. Ajoutez un zeste d'effets sonores. Quatre touches suffiront à vous envoyer dans le décor !

```
10 REM        **** CIRCUIT AUTOMOBILE ****
11 :
20 REM          AUTEUR : PATRICK CHOQUET
21
30 REM    COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
31 :
40 REM ***********************************************************
41 :
42 :
43 :
50 J=59464      : REM REGISTRE DU 6522
100 GOSUB 1020 : REM REGLE DE JEU
110 GOSUB 1220 : REM CONDITION INITIALE
120 GOSUB 1320 : REM DESSIN DU TERRAIN
140 GOSUB 1520 : REM ATTENTE D'UN CARACTERE
150 GOSUB 1720 : REM DEPLACEMENTS
155 GOSUB 20000: REM ACCIDENT
160 GOSUB 1920 : REM RESULTAT
161 :
170 PRINT : PRINT
180 PRINT TAB(5) "UNE AUTRE PARTIE !
185 PRINT : PRINT TAB(5) "DANS LES MEMES CONDITIONS O N?"
190 GOSUB 9020  REM ATTEND UNE REPONSE
195 :
200 IF R$="O" THEN GOSUB 11000 : C=0 : GOTO 140
210 R=0 :GOTO 110
220 :
230 :
240 :
1000 REM***********************************************************
1010 REM          REGLE DE JEU
1020 PRINT"'
1030 PRINT TAB(5)"ZZZZ CIRCUIT AUTOMOBILE ZZZZ"
1040 PRINT TAB(5)"++++++++++++++++++++++++++++"
1050 PRINT : PRINT : PRINT : PRINT"TAPEZ SUR 4 POUR ALLER VERS LA GAUCHE"
1070 PRINT : PRINT : PRINT              4 ------------------ DROITE"
1080 PRINT : PRINT : PRINT"             A ------------------ HAUT"
1090 PRINT : PRINT : PRINT"             Z ------------------ BAS"
1130 PRINT : PRINT : PRINT
1145 PRINT"              BONNE CHANCE !"
1147 GOSUB 10100   REM SON
1150 RETURN
1190 :
1191 :
1192
1200 REM ----------------------------------------------------------
1210 REM        CONDITION INITIALE
1220 PRINT : PRINT
1230 INPUT"VITESSE DE 1 A 100 KM/H ";V
1240 V=101-V
1250 IF V<1 OR V>100 THEN PRINT"", GOTO 1230 : REM ASCII 145 (CRSR HAUT)
1260 RETURN
```

Même sans permis, vous pourrez essayer le bolide sur l'un des circuits que vous propose votre CBM. Cependant, il vous faudra concentration, réflexes et toutes les autres qualités nécessaires à la conduite d'une véritable formule 1. Avant le départ, attachez votre ceinture, car après il sera trop tard.



En effet, vous vous en apercevrez très vite, la partie se termine généralement (à dire vrai à tous les coups) par un accident. Malgré tout, le but du jeu est de rester le plus longtemps possible sur la piste.

Le programme commence tout d'abord par vous indiquer comment vous déplacer. Ainsi, vous disposez des touches 4 et 6 et des lettres A et Z :

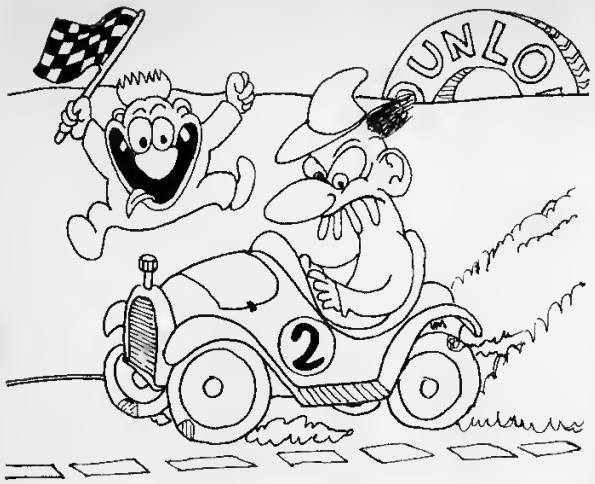
Ensuite, vous pourrez choisir votre vitesse initiale, mais attention, il n'y a pas de freins. Cependant, voici les notations :

1 conducteur saoûl .. (si vous

Première parution : L'OI n° 36

```
1290 :
1291 :
1292 :
1300 REM ----------------------------------------------------------------
1310 REM     PREPARATION DU TERRAIN
1320 PRINT"     : REM CLR
1330 C=0
1340 PRINT"  TAPER SUR UN NO POUR VISUALISER LES"
1350 PRINT   PRINT"            DIFFERENTS CIRCUITS" : PRINT : PRINT
1352 PRINT TAB(09)"1"TAB(19)"2"TAB(30)"3"
1355 PRINT . PRINT : PRINT : PRINT'      PUIS APPUYER SUR # QUAND VOUS"
1360 PRINT : PRINT"            AVEZ CHOISI LE CIRCUIT."
1362 .
1363 PRINT : PRINT : PRINT
1365 PRINT TAB(11) CHR$(18)'TAPER UN NUMERO'
1370 GOSUB 11000 : REM CHOIX DU TERRAIN
1375 :
1450 X=INT(1+40*RND(1)) : Y=INT(1+25*RND(1))
1460 IF PEEK(32767+(Y-1)*40+X)=46 THEN 1450
1470 RETURN
1480 :
1485 :
1490 :
1500 REM ----------------------------------------------------------------
1510 REM      ATTENTE D'UN CARACTERE
1520 Z=0
1530 GET R$
1540 Z=Z+1
1550 IF Z=15 THEN GOSUB 1620
1555 POKE J,0     : REM INITIALISE SON
1560 IF R$="  THEN 1530
1570 RETURN
1580 :
1585 :
1590 :
1600 REM ----------------------------------------------------------------
1610 REM  CLIGNOTEMENT DU SOLIDE
1620 Z=0 : X=13 : Y=23 : D=98
1630 POKE 32767+(Y-1)*40+X,D+128 : POKE J,99
1635 FOR I=1 TO 30 : NEXT I
1640 POKE 32767+(Y-1)*40+X,D : POKE J,200
1650 RETURN
1660 :
1670 :
1680 .
1700 REM ----------------------------------------------------------------
1710 REM          DEPLACEMENTS
1720 B=ABS(R$)
1730 IF B=52 THEN M=-1 : N=0 : D=93 : BID=B : GOTO 1770
1740 IF B=54 THEN M=1  : N=0 : D=93 : GOTO 1770
1750 IF B=65 THEN M=0  : N=-1: D=118: BID=B : GOTO 1770
1760 IF B=90 THEN M=0  : N=1 : D=114: GOTO 1770
1765 GOTO 1820
1767 :
1768 :
1770 X=X+M : Y=Y+N
1780 IF PEEK(32767+(Y-1)*40+X)=46 THEN 1840
1781 :
1782 :
1783 IF B=BID THEN 1790
1784 IF B>55 AND BID>55 THEN C=ABS(C-1)  GOTO 1800
1785 IF B<55 AND BID<55 THEN C=ABS(C-1)  GOTO 1800
1789 :
1790 C=C+1
1791 :
1800 POKE 32767+(Y-1)*40+X,D : POKE J,X+Y+20
1807 POKE 32767+((Y-N)-1)*40+(X-M),32
1809 :
1810 FOR T=1 TO V : NEXT T
1815 :
1820 GET R$ : POKE J,0
1830 IF R$="" THEN 1770
1835 GOTO 1720
1837 :
1838 :
1840 RETURN
1850 :
1860 :
1870 :
1900 REM ----------------------------------------------------------------
1910 REM      AFFICHAGE DU RESULTAT
1915 :
1920 FOR T=1 TO 1000 : NEXT T
1925 :
1930 PRINT'"     : REM CLR CB CB
1940 PRINT TAB(5) CHR$(18) 'VOTRE SCORE EST DE 'C' POINTS' : POKE J,0
1950 PRINT : PRINT
1960 IF C<=ZZ THEN PRINT'LE MEILLEUR SCORE EST TOUJOURS CELUI DE ' : GOTO 10000
1970 PRINT 'BRAVO. VOUS AVEZ LE MEILLEUR SCORE !!'
1980 PRINT : INPUT "QUEL EST VOTRE PRENOM ',N$
1990 ZZ=C
2000 RETURN
2010 :
2020 :
2030 :
9000 REM ----------------------------------------------------------------
9010 REM      TEST SUR LA REPONSE
9020 GET R$
9030 IF R$="" THEN 9020
9040 IF R$<>'O' AND R$<>'N' THEN PRINT :PRINT TAB(5) 'O OU N S.V.P' : :GOTO 9020
9050 RETURN
9060 :
9070 :
9080 :
9900 REM ----------------------------------------------------------------
10000 PRINT      "N$ . PRINT . PRINT'      AVEC 'Z7" POINTS.' : GOTO 2000
10010 :
10050 :
10060 REM   SON
10070 :
10100 POKE J+3,16 : POKE J+2,15
10105 :
10110 FOR I=100 TO 1 STEP -1
10120      POKE J,I
10130 NEXT I
10135 .
10140 POKE J,0
10150 RETURN
10160 :
10170 :
10180 .
10500 REM ----------------------------------------------------------------
10510 REM      CHOIX DU TERRAIN
11000 IF R=0 THEN 11020
11005 ON R GOSUB 11100,11130,12000
11010 RETURN
11015 .
11016 :
11017 :
11018 REM ----------------------------------------------------------------
```

```
11019 REM CHOIX DU CIRCUIT
11020 GETA$
11025 :
11030 IF A$='1' THEN R=1 : GOSUB 11100
11040 IF A$='2' THEN R=2 : GOSUB 11130
11050 IF A$='3' THEN R=3 : GOSUB 12000
11060 IF A$="#' AND R<>0 THEN RETURN
11065 .
11070 GOTO 11020
11075 :
11080 :
11085 :
11087 REM ----------------------------------------------------------------
11090 REM          DESSIN DES CIRCUITS
11095
11097 .
11100 PRINT"  .....  .....   ......................."
11101 PRINT"  "SPC(12)".  ."SPC(20)'.'
11102 PRINT"."SPC(14)"..'SPC(22)".';
11103 PRINT"."SPC(15)"."SPC(22)' ';
11104 PRINT".      ....     -   ................   .";
11105 PRINT".   .        .   .                    .  ";
11106 PRINT".   .        .   .                    .  ";
11107 PRINT".   .        .   ..                   .  ";
11108 PRINT".   .   .    .   ................   .  ";
11109 PRINT".   .   .    .   ..                   .  ";
11110 PRINT".   .   .    .   .                    .  ";
11111 PRINT".   .   .    .   .                    .  ";
11112 PRINT".   .   .    .   ..................   .  ";
11113 PRINT".   .   .    .   .                    .  ";
11114 PRINT'.   .   .    .   .                    .  ";
11115 PRINT".   .   .    .   ..                   .  ";
11116 PRINT".   .   .    .    ................   .  ";
11117 PRINT".   .        .                        .  ";
11118 PRINT'.   .   .    .                        .  ";
11119 PRINT".   .   .    .                        .  ";
11120 PRINT'.   .   .     ......................   .  ";
11121 PRINT".   .        .                        .  ";
11122 PRINT'.   .        .                        .  ";
11123 PRINT".   .    . .                          .  ";
11124 PRINT'  ....................................."
11125 RETURN
11126 :
11127 :
11128 :
11130 PRINT'  ..........  ........................  .
11131 PRINT" .          ..                        .  ";
11132 PRINT".           .                         .  ";
11133 PRINT".           .                         .  ";
11134 PRINT'.     . .   .     ................    .  ";
11135 PRINT".           .   .        .  .       .  ";
11136 PRINT".           .   .        .  .       .  ";
11137 PRINT".   .       .   .        .  .       .  ";
11138 PRINT'.   .     ..    .     ..   .  .     .  ";
11139 PRINT'.   .       .   .  .  .   .  .     .  ";
11140 PRINT".   .       .   .  .  .   .  .     .  ";
11141 PRINT".   .       .   .  .  .   .  .     .  ";
11142 PRINT'.   .       .   .  .  .   .  .     .  ";
11143 PRINT'.   .       .    ......    .  .     .  ";
11144 PRINT'.   .       .            .    .     .  ";
11145 PRINT".   .       .           .     .     .  ";
11146 PRINT'.   .     .......          . .     .  ";
11147 PRINT".   .                 .           .  ";
11148 PRINT".   .                 .           .  ";
11149 PRINT'.   .                ...          .  ";
11150 PRINT'.     ..............................  ";
11151 PRINT'.                                     .";
11152 PRINT'.                                     .";
11153 PRINT'.                                     .";
11154 PRINT"  ....................................."
11155 RETURN
11160 :
11170 .
11180 :
12000 PRINT"  ..............  ......................  ";
12001 PRINT'.                 .  .          .      .  ";
12002 PRINT'.                    ..         .      .  ";
12003 PRINT'.    ..  .....       .    ....  .      .  ";
12004 PRINT'.   .        .    .   .   .  .  .      .  ";
12005 PRINT'.   .        .    .   .   .  .  .      .  ";
12006 PRINT'.   .      ....   .   .   .  .  .      .  ";
12007 PRINT'.   .    .     .   .   .   .  .  .      .  ";
12008 PRINT'.   .    .     .   .   .   .  .  .      .  ";
12009 PRINT'.   .    .     .   .   .   .  .  .      .  ";
12010 PRINT'.   .    .     .   .   .   .  .  .      .  ";
12011 PRINT'.   .    .     .   ..  .   .  .  .      .  ";
12012 PRINT'.   .    .     .   .   .   ..  .  .      .  ";
12013 PRINT'.   .    .     .   . ..... .  .  .      .  ";
12014 PRINT'.   .    .     .   ..      ..  .  .      .  ";
12015 PRINT'.   .    .     .   .        .  .  .      .  ";
12016 PRINT'.   .    .     .   .        .  .  .      .  ";
12017 PRINT'.   .    .     .   .        .  .  .      .  ";
12018 PRINT'.   .    .     .   .        .  .  .      .  ";
12019 PRINT'.   .    .     .   .        .  .  .      .  ";
12020 PRINT'.   .      .   .   .      .   .  .      .  ";
12021 PRINT'.    ..............................      .  ";
12022 PRINT'.                                     .  ";
12023 PRINT'.                                     .  ";
12024 PRINT'  ....................................."
12025 RETURN
12050 :
12060 .
12070 :
19000 REM ----------------------------------------------------------------
19010 REM        ACCIDENT ET SONORISATION
19020 .
20000 FOR U=1 TO 5
20005 :
20010     G=128
20015 :
20020     FOR I=1 TO 2
20030         POKE 32767+(Y-2)*40+(X-1),77+G
20035         POKE 32767+(Y-2)*40+X,93+G
20040         POKE 32767+(Y-2)*40+(X+1),78+G
20050         POKE 32767+(Y-1)*40+(X-1),64+G
20070         POKE 32767+(Y-1)*40+X,42+G
20080         POKE 32767+(Y-1)*40+(X+1),64+G
20090         POKE 32767+Y*40+(X-1),78+G
20100         POKE 32767+Y*40+X,93+G
20110         POKE 32767+Y*40+(X+1),77+G
20115         POKE J,I+U+35
20120         G=G-128
20130     NEXT I
20135 :
20140 NEXT U
20145 :
20150 POKE J,0     : REM ARRETE LE SON
20155 :
20160 RETURN
READY.
```



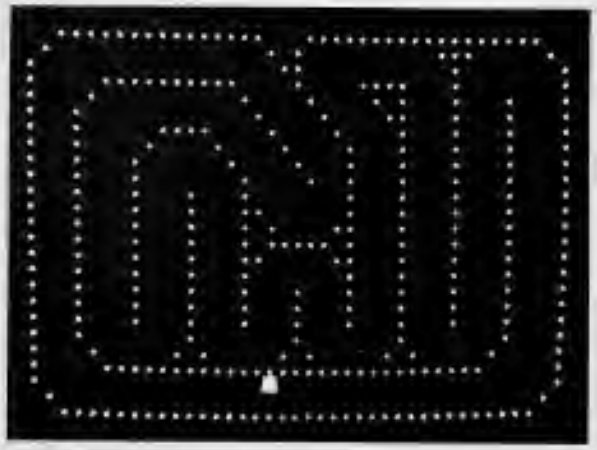
Remarques

• Sous cette forme le programme occupe 9 Ko Si vous supprimez les Rem et les espaces entre les mots-clefs ou même un des trois circuits vous serez nettement en dessous des 8 Ko

• Il est preferable d'utiliser une interface sonore ce qui ameliore le côte « realiste » de la course Mais si le son vous gêne vous pouvez le supprimer en tapant ligne 1980 apres la question « quel est votre prenom ? », une suite de caractères qui mordent sur la colonne 35 de votre CBM

• De meme si l'emplacement des touches clefs ne vous convient pas, agissez sur les lignes 1730 a 1760

• Enfin pour atteindre une vitesse supeneure a 100 km/h modifiez le programme aux lignes 1230 a 1250



voyez deux ordinateurs au lieu d'un)
25 si vous débutez
50 c'est déjà mieux
75 ça commence à foncer
100 à vos risques et périls

Le programme vous demande un numéro, vous avez le choix entre trois circuits (moyen, tres difficile, suicidaire) Quand vous avez trouvé celui qui vous plait appuyez sur # et vous verrez apparaître votre bolide Son moteur tourne et vous pouvez l'entendre, si vous avez, bien sûr, branché votre haut parleur Votre formule 1 n'attend plus maintenant que vos instructions pour partir Mais, accrochez-vous, car dès que vous rencontrerez une botte de paille votre véhicule se désintegrera purement et simplement

Enfin, si vous êtes le meilleur, l'ordinateur vous demandera quelques renseignements administratifs sinon, il vous narguera, en vous rappelant le score et le nom de l'as du volant qui a fait mieux que vous

Une derniere précision à chaque accident (toutes les 30 secondes environ), votre ordinateur vous demandera si vous voulez faire une dernière tentative sur le même circuit et à la même vitesse

Ce programme tient sur 9K, mais si votre CBM (3 000 ou 4 000) a une mémoire plus étendue, rien ne vous empêche de rajouter un autre circuit Pour cela, il suffit de modifier la ligne 11005 en ajoutant le numéro ou commencera le dessin de votre nouveau circuit, et d'autre part rajouter une instruction entre les lignes 11020 et 11060 pour que l'ordinateur sache que lorsque vous appuyez sur la touche 4, il faut afficher le nouveau circuit

Il ne me reste plus qu'à vous souhaiter bonne chance

Et bonne route

Patrick Choquet

alice au pays des merveilles logiques

En 1896, à Londres, le Révérend C.L. Dodgson, habile photographe, ennuyeux professeur de mathématiques et logicien passionné, publia un ouvrage intitulé « Symbolic Logic », traduit, bien plus tard, en français, sous le titre « Logique sans peine » (Hermann, 1966). Il y proposait certaines méthodes propres à déceler « la », ou « les » conclusions cachées dans une liste d'hypothèses.

Les exercices présentes, dans le droit fil d'Alice, jaillissaient d'univers fantastiques, liés cependant au nôtre par une commune logique

Voici un exemple de ses exercices

1) Aucun chaton qui aime le poisson n'est réfractaire à l'étude

2) Aucun chaton sans queue n'est prêt à jouer avec un gorille

3) Les chatons moustachus aiment toujours le poisson

4) Aucun chaton amoureux de l'étude n'a les yeux verts

5) Aucun chaton n'a de queue s'il n'est moustachu

Avez-vous, cher lecteur (ou lectrice), trouvé la conclusion oh combien logique, de ces cinq prémisses ? Si oui, vives félicitations ! Sinon une curiosité irrépressible vous incitera, peut être, à poursuivre

Les procédés de Lewis Carroll utilisaient un ensemble de diagrammes en forme de damiers, et des listes d'indices A cette epoque, vous vous en souvenez, les ordinateurs n'étaient pas nés et la logique mathématique hasardait ses premiers pas

Les temps ayant changé, je me suis demandé si, en 1982, un PSI (petit système informatique), convenablement éduqué, pouvait résoudre les énigmes, contemporaines du Pays des Merveilles La réponse est positive Le programme ci-dessous baptisé « Lewis Carroll », accepte les hypothèses, en langage presque naturel, sous la forme suivante

a) Classe générale chaton

b) Liste d'attributs ou proprietés

1) Chaton aux yeux verts
2) Chaton qui aime le poisson
3) Chaton qui a une queue
4) Chaton amoureux de l'étude
5) Chaton moustachu
6) Chaton prêt a jouer avec un gorille

c) Prémisses :

– aucun 2 n'est non 4,
– aucun non 3 n'est 6,
– tout 5 est 2,
– aucun 4 n'est 1,
– tout 3 est 5

Après quelques brefs instants de réflexion, la réponse apparaît à l'écran ou sur l'imprimante

« Aucun chaton aux yeux verts n'est prêt à jouer avec un gorille »

C'est le bon sens même ! Voyons maintenant la structure du programme

Quelques éléments d'analyse logique

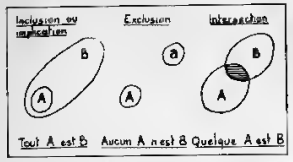
Définissons, d'abord, un « univers logique » Il décrit la classe générale des êtres ou objets qui seront considérés (choses, personnes, créatures vivantes, etc.)

Premiere parution : L'Or n° 41

Donnons-nous, ensuite, une liste d'attributs ou de propriétés de ces objets (avares, joyeux, qui attendent, riches aux yeux verts ...

A chaque propriété, peuvent être associés deux classes :
— celle des objets de l'univers logique qui ont cette propriété (classe C),
— la classe complémentaire (désignée par non C ou C) des objets qui ne jouissent pas de la propriété considérée.

Nous nous limiterons, ensuite, à des jugements ou propositions liant deux classes A et B. Ils ne peuvent être que de trois types, visualisés par des diagrammes :



En associant à toute classe C, la classe complémentaire C, l'exclusion peut être ramenée à l'implication. En effet :

« aucun A n'est B »

équivaut à :

« tout A est non B »

En définitive, dans le cadre, limité, des relations binaires, deux types seulement subsistent :
— l'implication ou inclusion,
— l'intersection.

Par ailleurs, à chacun de ces deux types, peuvent être toujours associées deux relations équivalentes, qui mettent en jeu les classes et leurs complémentaires.

Implication :
« tout A est B »,
« tout non B est non A ».

Intersection :
« quelque A est B »,
« quelque B est A »

Les déductions ou raisonnements ici considérés reposent sur deux types de chaînages ou transitivités entre relations :

a) Transitivité forte ou chaîne d'implications : si tout A est B, tout B est C, tout C est D, alors tout A est D. En bref, toute chaîne d'implications permet d'associer, par une implication finale (conclusion), les deux bouts de la chaîne.

b) Transitivité faible : si quelque A est B, tout B est C, tout C est D, alors quelque A est D. Autrement dit, toute suite de relations, commençant par une intersection et suivie d'une chaîne ininterrompue d'implications, permet de conclure à une relation d'intersection entre les deux extrémités de la chaîne.

Graphe des relations

A toute classe C, définie par une propriété ou un attribut, nous pouvons associer deux points d'un graphe représentant :
— l'un, la classe C elle-même,
— l'autre, la classe complémentaire C.

Toute relation binaire entre deux classes X et Y (ou leurs complémentaires) se traduira par deux arcs équivalents :

a) Implication : deux arcs en traits pleins

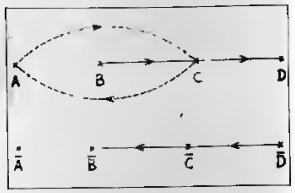
X ⟶ Y
Y ⟶ X

b) Intersection : deux arcs en traits pointillés

X ⇢ Y
Y ⇢ X

Exemple :

Le graphe représentatif des trois relations suivantes :
« quelque A est C »
« tout B est C »
« tout C est D »
apparaît ainsi :



Le graphe traduit, exactement, les prémisses. Reste à formuler une méthode systématique et sûre (un algorithme) pour trouver la ou les conclusions.

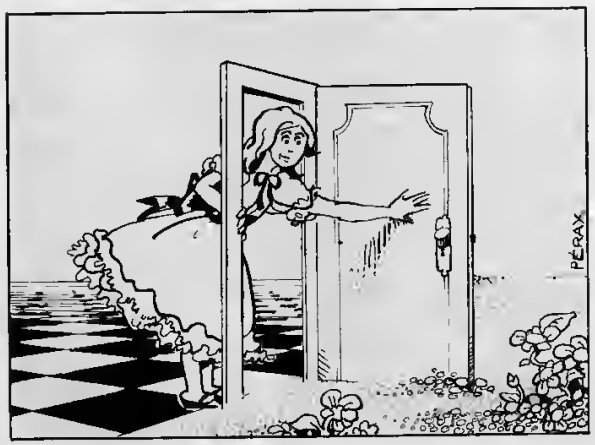
Pourrait être appelée « conclusion » toute relation nouvelle, symbolisée par un arc supplémentaire du graphe, créé en application des règles de transitivité (forte ou faible) plus haut énoncées.

De ce point de vue, les chaînes d'arcs
A ⟶ D ⟶ B
et A ⟶ C ⟶ E
entraîneraient
A ⟶ B
et A ⟶ E
quelles que soient les situations respectives des points dans le graphe. Ce jeu autoriserait, souvent, nombre de conclusions, certes vraies, mais sans intérêt majeur. Nous définirons le concept de conclusion, de façon plus restrictive, suivant, d'ailleurs, en cela, Lewis Carroll, qui distinguait des « éliminendes » et des « rétinendes ». Pour nous, une « conclusion » sera caractérisée comme suit :

a) relation nouvelle, donc non présente dans le graphe des prémisses ;

b) point terminal sans successeur à transitivité forte ;

c) point initial, au départ d'un arc « faible » ou sans ascendant, au départ d'un arc « fort ».




```
10 PRINT"";TAB(15)"LEWIS CARROLL"
20 PRINT PRINT"CE PROGRAMME TIRE LES CONCLUSIONS"
30 PRINT"D UN GROUPE INFINI DE PREMISSES"
40 DIM A$(28) OPEN 4,4
50 PRINT PRINT"ENTREE DES DONNEES"
60 PRINT INPUT"CLASSE GENERALE ";C$
65 PRINT#4,C$ PRINT#4
70 PRINT PRINT"LISTE DES ATTRIBUTS(TERMINEE PAR STOP)" PRINT
80 I=I+1 PRINT "-";" "; INPUT A$(I)
90 IF A$(I)="STOP" THEN 200
95 PRINT#4, I "-" A$(I)
100 GOTO 80
200 N=I-1 DIM H$(2N,2N),H1$(2N,2N),T$(2N),Z(2N)
210 PRINT#4 PRINT#4
250 REM*****************************
300 REM CONSTRUCTION DE LA TABLE DES RELATIONS
310 REM*****************************
320 PRINT PRINT"LISTE DES PREMISSES(TERMINEE PAR STOP)"
330 PRINT"TYPES D'ENTREE(X ET Y=NO.DES ATTRIBUTS)"
340 PRINT PRINT"TOUT X (OU NON X) EST Y (OU NON Y)"
350 PRINT"AUCUN X (OU NON X) N'EST Y (OU NON Y)"
360 PRINT"QUELQUE X (OU NON X) EST Y (OU NON Y)"
370 PRINT
380 K=K+1
390 PRINT"H";K;" "; INPUT H$
400 IF H$="STOP" THEN 620
402 REM*****************************
405 REM ANALYSE DE LA RELATION POSEE
408 REM*****************************
410 X$=H$ Y$="EST" GOSUB 2900
420 IF I=0 THEN GOSUB 2200 GOTO 390
430 H1$=LEFT$(H$,I-1) H2$=MID$(H$,I+3)
440 X$=H1$
450 Y$="TOUT" GOSUB 2800 IF I<0 THEN F$="" GOTO 490
460 Y$="AUCUN" GOSUB 2800 IF I<0 THEN F$="_" GOTO 490
470 Y$="QUELQUE" GOSUB 2800 IF I<0 THEN F$="*" GOTO 490
480 GOSUB 2200 GOTO 390
490 GOSUB 2100 IF I=0 THEN GOSUB 2300 GOTO 390
500 X=VAL(MID$(H$,1))
510 Y$="NON" GOSUB 2800 IF I<0 THEN Y=X+N
520 X$=H2$ GOSUB 2100 IF I=0 THEN GOSUB 2200 GOTO 390
530 Y=VAL(MID$(Y$,1))
540 Y$="NON" GOSUB 2800 IF I<0 THEN Y=Y+N
550 REM*****************************
555 REM CALCUL DE LA FORME CODEE D'UNE  RELATION
560 REM ET DE LA RELATION SYMETRIQUE
565 REM*****************************
570 GOSUB 4000
580 IF F$=" " THEN GOSUB 3000
590 IF F$="_" THEN GOSUB 3100
600 IF F$="*" THEN H$(X,Y)="*" H$(Y,X)="*"
610 PRINT#4,H$ GOTO 380
620 H=0 CN=0 PRINT
630 GOSUB 4300
1000 REM*****************************
1005 REM CALCUL DES CONCLUSIONS
1010 REM FERMETURE TRANSITIVE DU GRAPHE DES RELATIONS
1015 REM*****************************
1020 FOR J=1 TO 2N
1030 F=0
1040 FOR K=1 TO 2N
1050 IF H$(J,K)="*" THEN F=1
1060 NEXT K
1070 IF F=0 THEN T$(J)="A" GOSUB 4100
1080 NEXT J GOSUB 4600
1090 N=1 PRINT PRINT GOSUB 4200 PRINT
1092 REM*****************************
1095 REM ECRITURE DES CONCLUSIONS
1098 REM*****************************
1100 PRINTTAB(15)"CONCLUSIONS" PRINT
1110 FOR Y=1 TO 2N FOR Y=1 TO 2N
1120 GOSUB 4000
1130 IF H1$(X,Y)="*" OR X=Y THEN 1220
1135 IF H1$(X,Y)="T" AND YCXK THEN 1220
1140 IF H1$(X,Y)=" "AND )=YC THEN PRINT"HYPOTHESES CONTRADICTOIRES" GOTO1225
1145 IF H1$(X,Y)="Q" THEN 1150
1150 IF X<N AND Y<N THEN PRINT"TOUT ";C$;" ";A$(X);" EST ";A$(Y)
1160 IF X<N AND Y>N THEN PRINT"AUCUN ";C$;" ";A$(X);" N'EST ";A$(YC)
1170 IF X>N AND Y<N THEN PRINT"TOUT ";C$;" NON ";A$(XC);" EST ";A$(Y)
1180 IF X>N AND Y>N THEN PRINT"TOUT ";C$;" ";A$(XC);" EST ";A$(XC)
1185 GOTO 1220
1190 IF X<N AND Y<N THEN PRINT"QUELQUE ";C$;" ";A$(X);" EST ";A$(Y)
1195 IF X<N AND Y>N THEN PRINT"QUELQUE ";C$;" ";A$(X);" EST NON ";A$(YC)
1200 IF X>N AND Y<N THEN PRINT"QUELQUE ";C$;" NON ";A$(XC);" EST ";A$(Y)
1210 IF X>N AND Y>N THEN PRINT QUELQUE ";C$;" NON ";A$(XC);" EST NON ";A$(YC)
1220 NEXT Y,X
1225 PRINT#4 CLOSE4
1230 END
1500 REM
```

```
1900 REM ******SOUS PROGRAMMES*******
1905 REM
1910 REM RECHERCHE DE LA POSITION DANS X$ D'UNE CHAINE Y$
1915 REM
2010 IF Y$="" THEN I=0 RETURN
2020 FOR I=1 TO LEN(Y$)
2030 IF MID$(Y$,I,LEN(X$))=X$ THEN RETURN
2040 NEXT I I=0 RETURN
2100 REM
2105 REM RECHERCHE D'UN NOMBRE DANS LA  CHAINE X$
2108 REM
2110 IF X$="" THEN I=0 RETURN
2120 FOR I=1 TO LEN(X$)
2130 Y$=MID$(X$,I,1)
2140 IF ASC(Y$)<48 AND ASC(Y$)>57 THEN RETURN
2150 NEXT I I=0 RETURN
2200 REM
2205 REM IMPRESSION MESSAGE D'ERREUR
2208 REM
2210 PRINT PRINT ERREUR" PRINT
2220 RETURN
2230 REM
2240 REM CODAGE DES RELATIONS
2250 REM
3000 H$(Y,Y)="^" H$(YC,YC)="^" RETURN
3100 H$(X,YC)="^" H$(Y,XC)="^" RETURN
3110 REM
4000 REM CALCUL DES INDICES DES CLASSES COMPLEMENTAIRES
4010 REM
4020 IF Y<=N THEN YC=Y+N GOTO 4040
4030 YC=Y-N
4040 IF Y<=N THEN YC=Y+N RETURN
4050 YC=Y-N RETURN
4060 REM
4070 REM CALCUL DE LA MATRICE DES CONCLUSIONS
4080 REM
4100 K=0 F=0
4110 K=K+1 IF K>2N THEN 4220
4120 IF T$(K)<>"A" THEN 4110
4130 G=0
4150 FOR I=1 TO 2N
4160 IF H$(I,K)=" " THEN 4190
4170 IF H$(I,K)="^" THEN T$(J)="A" G=1 F=1 GOTO 4190
4180 IF H$(I,J)<>"*" THEN H1$(I,J)="Q"
4190 NEXT I T$(K)=""
4200 IF G=0 THEN IF N$(K,J)<>"^" THEN H1$(K,J)="T"
4210 GOTO 4110
4220 IF F=1 THEN 4100
4230 RETURN
4240 REM
4250 REM IMPRESSION DES MATRICES
4260 REM
4300 PRINT"  |" FOR J=1 TO 2N
4310 Z=J IF Z>N THEN Z=Z-N
4315 GOSUB 4500
4320 PRINT Z$ "|".
4330 NEXT J PRINT
4335 FOR J=1 TO 5N+3 PRINT"-";NEXT J PRINT
4340 FOR I=1 TO 2N
4350 Z=I IF Z>N THEN Z=Z-N
4355 GOSUB 4500
4360 PRINT Z$ "|"
4370 FOR J=1 TO 2N
4380 IF I=J THEN 4410
4390 IF H$(I,J)=" " THEN H$(I,J)=" "
4400 PRINT H$(I,J)+" |" NEXT J PRINT GOTO 4430
4410 IF H$(I,J)="" THEN H1$(I,J)=" "
4420 PRINT H1$(I,J)+" |" NEXT J PRINT
4430 NEXT I
4440 RETURN
4450 REM
4460 REM FORMATAGE DES INDICES
4470 REM
4500 Z$=STR$(Z) Z$=MID$(Z$,2)
4510 IF LEN(Z$)<3 THEN Z$=Z$+" "
4520 RETURN
4600 REM
4610 REM CONCLUSIONS COMPLEMENTAIRES
4620 REM
4630 FOR J=1 TO 2N FOR J=1 TO 2N Z(J)=0 NEXT J
4640 FOR J=1 TO 2N
4650 IF H1$(I,J)="T" THEN Z(J)=J
4660 NEXT J
4670 FOR J=1 TO 2N-1
4680 FOR K=J+1 TO 2N
4690 IF Z(J)=J AND Z(K)=K THEN N1$(J,K)="Q"
4700 NEXT K,J,I
4710 RETURN
READY.
```

L'algorithme de recherche des conclusions comprend donc les phases suivantes :

a) rechercher les sommets terminaux, c'est-à-dire, sans successeur lié par un arc « fort » ;

b) à partir de chaque sommet terminal, « remonter » le graphe, en déterminant tous les ascendants de ce sommet, jusqu'à rencontre des points sans ascendant ou des arcs faibles ;

c) écrire les transitions résultant des règles de transitivité, c'est-à-dire tracer les arcs nouveaux. C'est une forme du processus appelé par les mathématiciens « fermeture transitive du graphe » ;

d) éliminer les éventuelles « conclusions » de ce type déjà présentes dans le graphe initial.

Ainsi, les conclusions tirées du graphe très simple, pris, plus haut, en exemple, seraient :
sommet terminaux : A, Ā, B̄, D,
conclusions : $\bar{D} \longrightarrow \bar{B}$,
$\bar{A} \dashrightarrow \bar{D}$, $B \longrightarrow D$

Certaines conclusions, on le remarquera, sont équivalentes, par exemple ici : B̄ ⟶ D̄ et D ⟶ B. Nous n'en présenterons qu'une.

Notre ensemble de règles peut être complété sur un point particulier.

Les relations :
« tout A est C »
« tout A est D »
entraînent, à l'évidence, que les classes C et D ont une partie commune, savoir A. En conséquence, la relation « quelque C est D » est une conclusion valable, qui doit être ajoutée à la liste.

Le jeu, ici décrit, pourrait, en principe, nécessiter seulement un crayon et beaucoup de papier. Lewis Carroll l'aurait sûrement découvert, s'il avait connu la théorie des graphes. En pratique, pour un nombre même restreint de prémisses, la combinatoire des arcs devient vite pénible et, bien sûr, l'ordinateur arrive à point, pour nous aider.

Représenter données, et phases du programme

Les points importants, pour une bonne compréhension du programme, concernent la représentation des données et l'algorithme de calcul des conclusions.

Après indication de l'univers logique (ou classe générale) considéré(e), la liste des N attributs est entrée, sous forme d'une table A $ (I).

Le graphe logique, traduisant les hypothèses ou prémisses est, comme il se doit, transformé en une matrice H $ (I, J). Chaque attribut (de numéro X) crée deux classes complémentaires d'indices X (de 1 à N) et XC (de N+1 à 2 N).

Symboles utilisés :
H $ (I, J) = « ↑ » signifie « classe I incluse dans J »
H $ (I, J) = « * » signifie « classe I intersecte J »

Chaque prémisse engendre deux relations équivalentes de cette forme. Par exemple, s'il y a 10 attributs, l'hypothèse :
« tout 3 est 5 »
donne
H $ (3, 5) = « ↑ » et H $ (15, 13) = « ↑ »

La première partie du programme (instructions 10 à 630) réalise une analyse syntaxique des prémisses, afin de construire et d'écrire la matrice H $ équivalente (interpréteur très simplifié).

Une autre matrice H 1 $ (I, J) représente les conclusions, selon le codage suivant :
H 1 $ (X, Y) = « T » signifie « tout X est Y »
H 1 $ (X, Y) = « Q » signifie « quelque X est Y »

Les instructions 1100 à 1220 traduisent ces relations finales en un français approximatif, mais, je l'espère, compréhensible.

Le cœur du programme est l'algorithme de calcul des conclusions, ci-dessous détaillé (instructions 1000 à 1080 et sous-programme 4100). Il utilise le marquage, par un « A » de tous les ascendants d'un sommet donné.

A) POUR toute classe d'indice J de 1 à 2 + N

Si J a un successeur fort, ALLER en B)

SINON T $ (J) = « A » ALLER au sous-programme C)

B) SUIVANT de J

C) indice colonne K = 0 ; drapeau F = 0

D) K = K + 1 SI K > 2*N, ALLER en F)

SI T $ (K) # « A », ALLER en D)

SINON ; drapeau G = 0

POUR indice ligne I de 1 à 2 N

SI H $ (I, K) = « ↑ », ALLER en E)

SI H $ (I, K) = « ↑ », T $ (I) = « A » ; G = 1 ; F = 1 ALLER en E)

SINON H 1 $ (I, J) = « O »

E) SUIVANT de I

T $ (K) = « »

SI G = 0, H 1 $ (K, J) = « T »

ALLER en D)

F) SI F = 1, ALLER en C)

RETOUR

Liste des variables

A $ (I)	= table des attributs
H $ (I, J)	= table des symboles définissant les relations entre classes, selon les prémisses indiquées
H1 $ (I, J)	= table des symboles définissant les relations entre classes, selon les conclusions calculées
T (I)	= table intermédiaire des ascendants d'un sommet du graphe
Z (I)	= table intermédiaire des indices utilisés pour le calcul des conclusions complémentaires
C $	= classe générale
F G H	= drapeaux
F $	= type d'une relation entre deux classes
H $	= chaîne de caractère contenant une hypothèse ou prémisse
H1 $ H2 $	= fragments de la chaîne H $ analysée
N	= nombre des attributs
X Y XC YC	= indices de classes et de leurs complémentaires
X$ Y$	= variables intermédiaires des sous-programmes d'analyse de chaînes
Z Z$	= indices courants d'une classe et chaîne associée utilisés dans le sous-programme d'impression des matrices

Seuls les agents de police du secteur sont des poètes

Le programme « Lewis Carroll » a été écrit pour un CBM, muni d'une imprimante. Cela permet de garder trace des hypothèses, des conclusions et des matrices H $ (I, J) et H 1 $ (I, J).

Pour travailler sans imprimante, il convient d'apporter les modifications ci-dessous :

Instruction 40 : supprimer OPEN 4,4

Instruction 65 : supprimée

Instruction 95 : supprimée

Instruction 210 : supprimée

Instruction 610 : supprimer PRINT # 4, H $

Instruction 620 : supprimer CMD 4

Instruction 1225 : supprimer PRINT # 4

Afin d'exercer vos capacités logiques et de nourrir le programme, je vous propose l'énigme suivante, en provenance

directe de Lewis Carroll :

1. Tous les agents de police du secteur dînent avec notre cuisinière.

2. Aucun homme aux cheveux longs ne peut être autre chose que poète.

3. Amos Judd n'a jamais fait de séjour en prison.

4. Les « cousins » de notre cuisinière aiment tous le gigot froid.

5. Seuls les agents de police du secteur sont poètes.

6. Seuls ses « cousins » dînent avec notre cuisinière.

7. Les hommes aux cheveux courts ont tous fait un séjour en prison.

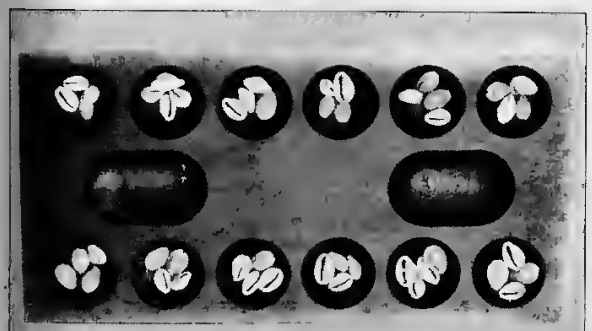
Conclusion ?

Bonne chance !

René Descamps

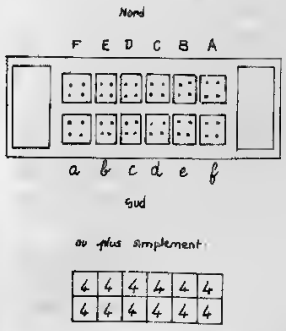
et si on jouait à l'Awalé sur TRS 80 ?

L'Awalé, bien que peu connu en Europe, est l'un des jeux les plus répandus dans le monde. Il se pratique dans presque toute l'Afrique. On le retrouve sous divers noms et parfois sous différentes variantes. C'est la variante du pays Baoule (Côte d'Ivoire) qui vous est proposée ici, accompagnée d'une liste en Basic, pour TRS-80. Bien que de réglementation souple, ce jeu vous permettra de développer des trésors d'intelligence tactique et les stratégies les plus élaborées.



Le jeu d'Awalé est connu en Afrique sous différentes appellations et il en existe énormément de variantes. On parle ainsi d'Awélé au Sénégal (la planche possède alors deux rangées de six cases), de Gissono au Burundi ou de Bao en Tanzanie. Dans ces deux derniers jeux, la planche comporte alors quatre rangées de huit cases. A remarquer que ces jeux, qui sont à l'Afrique ce que les échecs sont à l'Europe, ont été créés initialement à l'ouest de l'Afrique sous une forme simple et que règles et damier se sont progressivement sophistiqués en suivant le sens des grandes migrations.

Le matériel est très sobre : un « tablier », généralement une planchette de bois, dans laquelle ont été sculptées quelques alvéoles (le plus souvent douze pour le jeu lui-même, quelquefois deux de plus pour amasser les gains), et des « graines » qui peuvent être des grains de café, des haricots ou simplement des cailloux, au nombre de quarante-huit.



Ce jeu se pratique à deux.

Chaque rangée de six alvéoles représente le camp de chaque joueur, que nous appellerons Nord et Sud.

En début de partie, chaque case contient quatre graines.

Le premier joueur (Nord ou Sud) peut être tiré au sort ou laissé au choix des adversaires.

Première parution : L'OI n° 43

Les règles sont également très simples.

1) Sens du jeu : le jeu se déroule dans le sens inverse de celui des aiguilles d'une montre.

2) Déroulement de la partie : les adversaires jouent alternativement, chacun prend à son tour toutes les graines contenues dans l'une quelconque de ses cases et les sème, une par une, dans le sens indiqué (on parlera de « semis »).

Si la case jouée contient plus de onze graines, le joueur effectue un second tour, en sautant sa case de départ.

Des grains de café, des haricots ou des cailloux, une question de goût

3) Prise : si, lors d'un semis, un joueur dépose sa dernière graine dans une case adverse contenant déjà une ou deux graines, il prend alors tout le contenu de cette case (deux ou trois graines) et le dépose dans la case réservée à ses grains.

Si la case précédant celle de la prise est dans le camp adverse et contient deux ou trois graines, ces dernières sont également prises, et ainsi de suite, jusqu'à ce que l'on se trouve en présence d'une case nous appartenant en propre ou vide ou contenant plus de trois graines (ou que cette prise nous place dans le cas 4, auquel cas il faudra jouer une autre case).

4) Nécessité de donner à « manger » : il peut arriver, en cours de partie, qu'un joueur possède toutes les graines de son côté. Il doit alors (s'il le peut) jouer une case donnant la possibilité à son adversaire de jouer à son tour. En cas d'oubli de cette règle de courtoisie, il conviendra de la rappeler, sans sanction, au joueur « étourdi » qui devra recommencer son coup.

Cette règle s'appliquera de même lorsqu'une prise éventuelle démunit totalement l'« adversaire » : on renoncera alors à celle-ci.

5) Fin de partie : dès que l'un des joueurs totalise un gain supérieur à 24, c'est lui le vainqueur.

Si l'un des joueurs est dans l'impossibilité de jouer (toutes ses cases sont vides), il reçoit alors une graine, son adversaire ajoutant à ses gains les graines restant dans son camp (le vainqueur est aussi celui possédant le plus de graines).

```
10 CLS : CLEAR 200 : ED
20 PRINTTAB(5) " * * *
```

La défense contre une menace simple

4	·	4	2	1	·
·	·	6	·	·	

La case c du Sud menace de prendre les graines des cases B et C de Nord. Que peut faire Nord pour l'en empêcher ? Il y a trois possibilités (au moins).

a) Jouer C : ainsi c ne menace plus qu'une case vide et ne prendra rien. (Nord a bien sûr maintenant une graine de plus en E. Le total de E étant sans importance par rapport au problème).

4	·	5	0	1	·
·	·	6	·	·	·

b) Jouer B : ainsi c menace une case contenant déjà trois graines et ne prendra rien.

4	·	4	3	0	·
·		6	·	·	·

c) Jouer F : ainsi c ne menace plus C mais D qui, elle, contient quatre graines et ne prendra rien avec c.

0	·	4	2	1	·
·	·	7	·	·	·

Ce choix dépendra bien sûr des cases marquées « . » et de leur contenu.

		Case visée					
		e	b	c	d	e	f
Position de la maison	F	12	13	14	15	16	17
	E	13	14	15	16	17	18
	D	14	15	16	17	18	19
	C	15	16	17	1B	19	20
	B	16	17	18	19	1D	21
	A	17	18	19	20	21	22
		Nombre de graines dans la maison					

Le défense contre une menace multiple

La case c menace de prendre la graine D. Le case e menace de prendre la graine B. Nord ne peut alors empêcher Sud de prendre deux graines au coup suivant que s'il possède dans une des cases marquées « . » un nombre de graines suffisant pour effectuer la dernière manœuvre décrite (ajouter des graines dans les cases de l'adversaire).

·	·	1	·	1	·
·	·	7	·	3	·

Le défense contre une tenaille

La case e menace de prendre la graine A. Si Nord joue cette case A, Sud jouera f et prendra quand même deux graines. Nord ne peut alors empêcher Sud de prendre deux graines au coup suivant que s'il possède dans l'une des cases marquées « . » un nombre de graines suffisant pour effectuer le manœuvre précédente sans toutefois arriver jusqu'en B. Cherchez pourquoi.

·	·	·	·	0	1
·	·	·	·	2	2

L'attaque

L'attaque la plus simple consiste à se trouver soi-même dans l'une des positions décrites précédemment alors que l'adversaire ne peut pas se défendre. Il apparaît clairement, j'espère, que les menaces simples seront celles qui auront le moins de chances de réussir.

Une attaque efficace : la construction d'une maison

Construire une maison, c'est accumuler suffisamment de graines dans une case pour affectuer plus d'un tour de tablier, et gagner ainsi plusieurs ensembles de deux ou trois graines en un seul tour de jeu. La maison doit donc contenir un nombre minimal de graines : ce nombre dépand d'une part de la case dans laquelle on construit, d'autre part des cases visées.

Exemple : une maison, construite par Nord en D réclame deux graines pour arriver dans la dernière case plus six graines pour parcourir le camp de Sud, plus cinq graines pour parcourir le camp de Nord (on saute la case de départ D) plus au moins une graine pour atteindre à nouveau Sud en e, plus éventuellement une, deux, trois, quatre ou cinq graines pour atteindre b, c, d, e, f.

Attention : une de plus et Nord retombe chez lui. Mais la récolte peut être fructueuse.

16	4	2	1	0	3
1	2	0	1	1	3

0	5	3	2	1	4
3	4	0	0	0	4

Nord joue F et gagne trois graines en e, trois graines en d, deux graines en c soit au total huit graines !

Le tableau ci-dessus vous donne, en fonction de la case dans laquelle elle est construite, le nombre de graines nécessaires pour atteindre la case visée.

Maintenant à vous de jouer sur votre TRS-80. Mais peut-être connaissez-vous d'autres variantes. Alors, n'hésitez pas ! Envoyez-nous votre programme réalisé sur votre ordinateur individuel.

Jean-Louis Richaud

des oracles chinois sur votre ZX-81

Il ne se passe pas un jour sans que l'on ne soit amené à prendre des décisions plus ou moins importantes pour l'avenir. Parfois, entre deux possibilités, le choix est difficile. Votre ZX-81 peut alors vous secourir efficacement avec le programme que nous vous proposons et une pièce de monnaie (chinoise, du XIe siècle av. J.-C.).

Les sages chinois du troisième millénaire avant J.-C. avaient créé, pour se tirer d'embarras, un système d'oracles appelé *Livre des changements* ou *I Ching* (prononcer I King). Les bases de ces oracles furent établies en 3322 avant J.-C., par le saga Fu-Hsi.

Au XIe siècle avant J.-C., pendant la dynastie des Yin, le prince Wen et son fils, le duc de Tan, ont commenté le livre des changements. Plus tard, Confucius lui-même a longuement étudié le I Ching pour en tirer des enseignements profitables.

Les sages chinois étaient-ils des informaticiens ?

Le livre est composé de soixante-quatre signes constituant la réponse à une question. Afin de résoudre tous les problèmes, les réponses sont formulées de façon plus ou moins sybilline.

Pour se servir du fameux livre, les Chinois utilisaient un rituel compliqué. Dans le but de simplifier, vous utiliserez le système du pile ou face. En effet, le I Ching est fondé sur le principe du système binaire et fonctionne donc comme un ordinateur, qui n'utilise que le 1 ou le 0. Les Chinois se contentaient d'un simple trait interrompu (—) pour le 1 et d'un trait simple (–) pour le 0 ; On peut donc dire qu'au XXXIVe siècle avant J.-C., certains sages

Première parution : L'OI n° 47

Programme I Ching (début)

```
  1 REM I CMING
  2 REM ************************
  3 REM AUTEURS VILLIAM MARI ET
        DENIS OUMA
  4 REM ************************
  5 PRINT TAB 12;"[illegible]"
  6 PRINT
 10 PRINT "POUR APPORTER UNE SC
LUTION A  VOTRE PROBLEME, FAIT
ES PILE OU  FACE 6 FOIS EN VOUS
CONCENTRANT SUR LA QUESTION ET I
NSCRIVEZ VOSRESULTATS A CHAQUE T
OUR.","PILE = 1              F
ACE = 0"
 11 PAUSE 800
 12 POKE 16437,255
 13 CLS
 15 PRINT "PILE=1";TAB 26,"FACE
=0"
 25 LET A=10
 30 LET C$=""
 35 FOR N=1 TO 6
 40 LET P$=INKEY$
 41 IF R$<>"1" AND R$<>"0" THEN
GOTO 40
 45 IF R$="1" THEN PRINT AT A,J
3,"[illegible]"
 50 IF R$="0" THEN PRINT AT A,1
3;"[illegible]"
 55 LET C$=C$+R$
 60 LET A=A-2
 65 IF A=6 THEN GOSUB 80
 70 NEXT N
 75 GOTO 105
 80 IF C$="11" THEN PPINT AT 12
,0;"[illegible] C.EST L.EMBLEME D
E LAMERE, DE LA FEMINITE ACCUEIL
LAN-TE, DE LA DOUCEUR."
 85 IF C$="01" THEN PRINT AT 12
,0;"[illegible] C.EST L.EMBLEME D
E LAFILLE, DE LA JEUNESSE FEMINI
NE, DE LA RESERVE."
 90 IF C$="00" THEN PRINT AT 12
,0;"[illegible] C.EST L.EMBLEME
DU  PERE,DE LA VITALITE SAGE, DE
 LA FORCE."
 95 IF C$="10" THEN PRINT AT 12
,0;"[illegible] C.EST L.EMBLEME
DU  FILS, DE LA JEUNESSE VIRILE,
 DE L.IMPETUDSITE."
 100 RETURN
 105 PRINT
 110 IF C$="111111" THEN PRINT A
T 16,0;"[illegible] SUCCES PAR
L.OBEIS-SANCE."
 115 IF C$="111110" THEN PRINT A
T 16,0;"[illegible] NE BOUGEZ
 SURTOUTPAS."
 120 IF C$="111101" THEN PRINT A
T 16,0;"[illegible]: ETES-VOUS 6IEN SU
R DE  VOUS ?"
 125 IF C$="111100" THEN PRINT A
T 16,0;"[illegible]: TACHE INA
CHEVEE."
 130 IF C$="111011" THEN PRINT A
T 16,0;"[illegible]. RESPECT ENGEND
RE RES-  PECT."
 135 IF C$="111010" THEN PRINT A
T 16,0,"[illegible]: EXHIBEZ VOS QUA
LITES."
 140 IF C$="111001" THEN PRINT A
T 16,0,"[illegible] LE GRAND AIDE L
E PETIT."
 145 IF C$="111000" THEN PRINT A
T 16,0;"[illegible]  MANQUE DE CO
MMUNICA-TION."
 150 IF C$="110111" THEN PRINT A
T 16,0,"[illegible]  BRILLER SANS E
BLOUIR."
 155 IF C$="110110" THEN PRINT A
T 16,0,"[illegible]  REPOS ET CO
NCENTRA-TION."
 160 IF C$="110101" THEN PRINT A
T 16,0,"[illegible].CULTURE D.U
N TALENT"
 165 IF C$="110100" THEN PRINT A
T 16,0,"[illegible].FERMETE ET CO
RRECTION"
 170 IF C$="110011" THEN PRINT A
T 16,0,"[illegible]. SUCCES DANS
 LES  BROUILLES."
 175 IF C$="110010" THEN PRINT A
T 16,0,AT 16,0,"[illegible]  REC
MERCME D.UN  ACCORD."
 180 IF C$="110001" THEN PRINT A
T 16,0;"[illegible]: SAGACITE DU C
ONSEIL."
 185 IF C$="110000" THEN PRINT A
T 16,0;"[illegible] LA DIGNITE AVAN
T TOUT."
 190 IF C$="101111" THEN PRINT A
T 16,0;"[illegible]: SAGESSE DE
L.EXPERIENCE."
 195 IF C$="101110" THEN PRINT A
T 16,0,"[illegible]: INUTILE DE
 FAIRE  REPETER."
 200 IF C$="101101" THEN PRINT A
T 16,0;"[illegible]:LA SINCERITE EST
 PREFERABLE."
 205 IF C$="101100" THEN PRINT A
T 16,0,"[illegible].SUITE DANS LE
S IDEES."
 210 IF C$="101011" THEN PRINT A
T 16,0,"[illegible] REPRENNEZ L.A
NCIENNE FACON."
 215 IF C$="101010" THEN PRINT A
T 16,0,"[illegible]: LES ENNUIS
NE SONT PAS TERMINES."
 220 IF C$="101001" THEN PRINT A
T 16,0,"[illegible].POURQUOI TANT
 DE PAROLES ?"
 225 IF C$="101000" THEN PRINT A
T 16,0,"[illegible]: SAVOIR S.ARRETE
R EN CHEMIN."
 230 IF C$="100111" THEN PRINT A
T 16,0,"[illegible]: LE SAGE SAISI
T L.OCCASION."
 235 IF C$="100110" THEN PRINT A
T 16,0,"[illegible]: EXAMINEZ B
IEN LES CONSEQUENCES."
 240 IF C$="100101" THEN PRINT A
T 16,0;"[illegible]: ENTRAIDE ET CO
MPREMEN-SION."
 245 IF C$="100100" THEN PRINT A
T 16,0,"[illegible]:REPRENDRE PL
USIEURS FOIS."
 250 IF C$="100011" THEN PRINT A
T 16,0 "[illegible].TOUTES VOIE
S OUVER-TES."
 255 IF C$="100010" THEN PRINT A
T 16,0,"[illegible]. L.DREILLE ET L.
OEIL OU-VETS."
 260 IF C$="100001" THEN PRINT A
T 16,0,"[illegible]: INDIFFERENC
ES AUX  CANCANS."
 265 IF C$="100000" THEN PRINT A
T 16,0,"[illegible] SE MEFIER DU F
LUS FORT"
 270 IF C$="011111" THEN PRINT A
T 16,0,"[illegible].OBSTACLE SANS IMP
ORTANCE."
 275 IF C$="011110" THEN PRINT A
T 16,0;"[illegible] LA PENSEE ACCROI
T L.AC- TION."
 280 IF C$="011101" THEN PRINT A
T 16,0,"[illegible] TROP D.E
SPOIR  LA DECEPTION."
 285 IF C$="011100" THEN PRINT A
T 16,0,"[illegible].AVANCEZ SAN
S HESI- TER."
 290 IF C$="011011" THEN PRINT A
T 16,0,"[illegible] PAS D.APPREH
ENSION."
 295 IF C$="011010" THEN PRINT A
T 16,0,"[illegible].LA LOI PROTEG
E LE SA-GE."
 300 IF C$="011001" THEN PRINT A
T 16,0,"[illegible] UN GRAND PROTE
GE PAR  DES PETITS."
 305 IF C$="011000" THEN PRINT A
T 16,0,"[illegible] L.INCORRECTION
 MENE AUDESASTRE."
```

▶▶▶

▶▶▶

Programme I Ching (suite et fin)

```
 310 IF C$="010111" THEN PRINT A
T 16,0;"[illegible] POURQUOI
 ATTIREL.ATTENTION ?"
 315 IF C$="010110" THEN PRINT A
T 16,0,"[illegible] FIEZ-VOUS PAR
FDIS AUXAPPARENCES."
 320 IF C$="010101" THEN PRINT A
T 16,0;"[illegible] TERMINER EST
LE PLUS  HAL-AISE."
 325 IF C$="010100" THEN PRINT A
T 16,0,"[illegible] CHAQUE CHOSE
A SA PLACE."
 330 IF C$="010011" THEN PRINT A
T 16,0,"[illegible] INSUFFISANCE D
U DEMI- SUCCES."
 335 IF C$="010010" THEN PRINT A
T 16,0,"[illegible] LE NECESSAIRE DOI
T ETRE  FAIT."
 340 IF C$="010001" THEN PRINT A
T 16,0,"[illegible] AGIR D.ABORD,
 PENSER ENSUITE."
 345 IF C$="010000" THEN PRINT A
T 16,0,"[illegible] NE MELANC
EZ PAS  L.IMMISCIBLE."
 350 IF C$="001111" THEN PRINT A
T 16,0;"[illegible] SUCCES PEUT ETR
E PROVI-SOIRE."
 355 IF C$="001110" THEN PRINT A
T 16,0;"[illegible] L.EQUILIBRE
 DOIT  ETRE RETABLI."
 360 IF C$="001101" THEN PRINT A
T 16,0,"[illegible] FREIN DESSER
RE TOT  OU TARD."
 365 IF C$="001100" THEN PRINT A
T 16,0, "[illegible] LE SAGE CROIT D
ELUI QUICROIT."
 370 IF C$="001011" THEN PRINT A
T 16,0,"[illegible] VERTU DE L.AB
STENTION"
 375 IF C$="001010" THEN PRINT A
T 16,0,"[illegible]:LE MAUVAIS PRES
AGE SE  TROMPE."
 380 IF C$="001001" THEN PRINT A
T 16,0,"[illegible] NE PAS NEGLIGER LE
 PLAISIR."
 385 IF C$="001000" THEN PRINT A
T 16,0;"[illegible] MAIS LE TIGRE N
E MORD  PEUT ETRE PAS."
 390 IF C$="000111" THEN PRINT A
T 16,0,"[illegible] FIN DE PETITESSE, D
EBUT DE GRANDEUR."
 395 IF C$="000110" THEN PRINT A
T 16,0,"[illegible] S.INSPIRER DES
AUTRES."
 400 IF C$="000101" THEN PRINT A
T 16,0,"[illegible] NE PAS SE LAISSE
R CONDUIRE."
 405 IF C$="000100" THEN PRINT A
T 16,0,"[illegible] LE NUAGE N
E CREVE-RA PEUT ETRE PAS."
 410 IF C$="000011" THEN PRINT A
T 16,0,"[illegible].TRIOMPHE DE LA
 FERMETE"
 415 IF C$="000010" THEN PRINT A
T 16,0,"[illegible] TENEZ-VOUS BI
EN TOUS LES FILS EN MAIN."
 420 IF C$="000001" THEN PRINT A
T 16,0,"[illegible] REPANDEZ, N.AMASS
EZ PAS."
 425 IF C$="000000" THEN PRINT A
T 16,0,"[illegible] LE CHANGEMENT E
ST SALU-TAIRE."
 430 PRINT AT 21,0,"AVEZ-VOUS UN
E AUTRE QUESTION ?"
 435 INPUT P$
 440 IF P$="OUI" THEN GOTO 13
 445 CLS
 450 PRINT AT 12,10,"A BIENTOT"
 460 STOP
```

chinois parlaient comme nos cerveaux électroniques modernes !

Bien que fastidieux à entrer par le clavier du ZX-81 du fait de sa longueur, la structure de ce programme reste des plus simples.

Le traditionnel pile ou face sera déterminant

Les variables de la liste sont appelées suivant le code tapé par l'utilisateur. Elles donnent accès à l'impression d'un petit texte sur l'écran. Le calcul proprement dit se résume aux lignes 30 à 75, qui permettent de distinguer les données et d'afficher le symbole correspondant.

Pour utiliser ce programme, munissez-vous avant tout d'une pièce de monnaie.

Après avoir tapé RUN, le ZX-81 vous explique la règle du jeu. On lance la pièce ; si elle tombe sur pile on entre 1, si elle tombe sur face on entre 0. Cette opération est répétée cinq fois de suite.

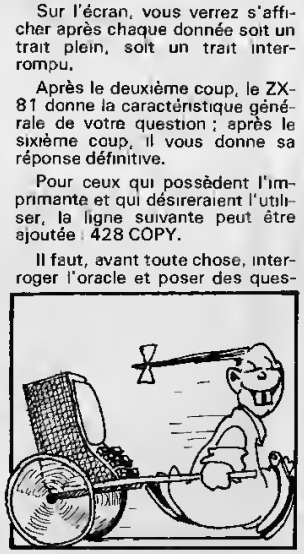
Grâce à la ligne 41, on ne peut qu'entrer 1 ou 0 sur le clavier ; toute action sur une autre touche restera sans effet, afin d'éviter les erreurs.

Sur l'écran, vous verrez s'afficher après chaque donnée soit un trait plein, soit un trait interrompu.

Après le deuxième coup, le ZX-81 donne la caractéristique générale de votre question ; après le sixième coup, il vous donne sa réponse définitive.

Pour ceux qui possèdent l'imprimante et qui désireraient l'utiliser, la ligne suivante peut être ajoutée : 428 COPY.

Il faut, avant toute chose, interroger l'oracle et poser des questions uniquement sur les événements contrôlables (hésitation sur le choix d'un métier, d'un achat, d'un spectacle, etc.).



Le I Ching ne vous fera malheureusement pas gagner au loto ou au tiercé.

Concentrez-vous sur la question qui vous préoccupe et lancez la pièce.

Si vous faites pile, tapez 1 ; si vous faites face, tapez 0 et ce jusqu'à ce que vous ayez lancé la pièce dix fois.

Un programme vieux de 5 300 ans !

Les réponses fournies ne sont pas toujours très claires mais voilà bien le propre des oracles. La réponse à votre question se trouve en vous, le I Ching vous aidera seulement à la découvrir ; c'est, en quelque sorte, un programme d'aide à la décision vieux de 5 300 ans.

Denis Duhr
William Mari

quand l'Atom a rendez-vous avec la Lune

Alunir ou atterrir ? Question d'atmosphère. Ce programme sur Atom doit vous aider à retrouver vos esprits, à faire le vide autour de vous et à vous concentrer sur un seul objectif : mener à bien votre mission. Mais attention, n'essayez pas de planer, car après tout, s'écraser sur la Lune (ou sur la Terre) ça fait tout de même très mal...

Le programme que nous vous proposons, entièrement graphique, vous transporte sur la Lune. Un module lunaire parcourt le haut de l'écran. Il est animé d'une vitesse horizontale aléatoire, et est soumis à la gravité lunaire.

Le joueur dispose de trois touches REPT-SHIFT-CONTROL. Les deux premières lui permettent d'allumer des rétrofusées pour se propulser respectivement à gauche et à droite. La troisième touche met en marche le réacteur principal : le module lunaire décolle alors vers les espaces interstellaires.

Bien sûr, les lois de l'inertie sont respectées, autrement dit l'accélération du module est contrôlée ; par exemple, si on appuie sur SHIFT le module accélèrera vers la gauche et si on lâche SHIFT, il gardera la vitesse acquise.

Pour l'arrêter, il faudra appuyer sur REPT, ce qui le fera décélérer.

Voilà ! Petit détail amusant : une fois devenu expert en maniement du module, essayez, avec la flamme des réacteurs de faire des trous dans le sol !

Ce programme est articulé suivant cinq étapes (les trois dernières étant réservées au programme secondaire), décrites en lignes Rem dans le programme.

Il ne vous reste qu'à essayer ce programme et à modifier la temporisation (ligne 460) suivant qu'il vous paraît trop simple ou trop compliqué. A bientôt peut-être, dans la Lune !

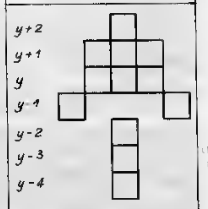
Francis Reignat

Première parution : L'Ordi n° 18

Programme d'alunissage (début)

```
 50  CLEAR0 P.$30              ALUNISSAGE
 60  P.'                     #15%207
 70  P.'                     $190%1E0$192'
 80  P.' COMMANDES "  '   CTRL  GAUCHE'' '  SHIFT  DROITE'
 90  P."   REPT  RETRO-FUSEE'' '   TAPER UNE TOUCHE ALIN#FFD3
100  DIM (0 130)
110  Y=96 H=P.%10 U=0.V=0
120 REM calcul du decor
130  CLEAR2
140  F.T=0TO130 Y%(T)=0 N.T
150 REM decor avant la base (calcul)
160  IF 0<=A.R.%20
170  F.X=0TO27
180 YY%=Y%+1 =Y%(X)>HR.%2
190  IF Y%(X+1)<2 G.9
200  N.X
210 REM dessin de la base (calcul)
220  F.X=25TO56 Y%(X+1)=Y%(X-25) N.X
230 REM dessin du decor apres la base
240  F.X=36TO128
250h YY%=Y%+1 =YY%+P.%2
260  IF YY%(X+1)<2 G.n
270  N.X
280 REM dessin effectif de l'ecr
290  F.X=0TO127 MOVE X,0 WAIT DRAW X,Y%(X) N.X '=64
300 REM dessin du cadre et soulignement de la base
310  MOVE0,50,PLOT7,0,0 PLOT7 127,0 PLOT7 127 50
320  MOVE26 YY%(25)-1 PLOT7(26,YY%(28)-1)
330 REM sequence Principale du Prog.
340 REM detection du module
350d M=15 GOS.f A=1
360 REM detection de la Position des touches
370  IF ?&B#FC&#A0=0 A=y+2 A=0
380  IF ?&F001&#20=0 H=H+1
390  IF ?&D001=#DF   H=H-1
400 REM calcul de la Position du module
410  M=M-1,Y=Y+V,X=X+H
420  IF X<1 X=2
430  IF X>126 X=126
440  M=1 GOS.f
450 REM temporisation
460  F.T=0TO8 WAIT N.T
```

Dessin du module



Copyright L'Ordi et l'auteur

Programme d'alunissage (fin)

```
470 REM crash ou alunissage?
480  IF X>27 IF X<37 IF Y<=3 Y%(X) G.r
490  IF (Y=2 Y%(X-2)OR Y=2 Y%(X+2) G.e
500 REM agrandissement?
510  IF U=0 IF Y<=Y%(32)<25 IF X>27 IF X<37 GOS.V
520  G.d
530 REM fin de sequence Principale
540 REM dessin d'explosion + commentaires
550e IF V<-4 CLEAR0 P.$20      APRES CET ALUNISSAGE DE" E=0
560  IF V<-4 P." FORTUNEA  (A.(32-5)) " MAIS DE LA PAGE IL NE "
570  IF V<-4 P."VOUS" 'RESTE PLUS QU'A MARCHER..."' G.1
580  F.T=0TO80 MOVE X,Y%(T),DRAW P.'40+X,Y%(A.P.'50+X,Y%(T))
590  ?&600D=?&6002 4,N.T
600  F.T=0TO50 WAIT N.T CLEAR0 P.$30 ''''
610  P.'''"   APRES CE CRASH MEMORABLE."
620  P."VOTRE MODULE EST INUTILISABLE!"'' LA BASE EPARGNEE "
630  F."OUS REND LES HONNEURS..." '
640  G.i
650 REM dessin du module
660f PLOTM (X-1),Y%(X+1) PLOTM (X+1)
670  PLOTM (X-1),Y%(X+1),PLOTM,Y%(X+1),PLOTM,Y%+1,Y%(X+1)
680  PLOT M,X,(Y+2),PLOT M,X-2,(Y-1) PLOT M,X+2,(Y-1)
690  IFA=0 PLOT M,X,(Y-2) PLOTM,X,(Y-3),PLOTM,X,(Y-4) R.
700  RETURN
710 REM commentaires
720r IF V<-4 CLEAR0 P.$30                  ********'
730  IF V>-4 P.            # BRAVO #'   '
740  IF V>-4 P.            ********'   '
750  IF V<-4 P.        SUPERBE ALUNISSAGE!"'' G.1
760  F.T=0TO30 MOVEY,Y%(X) DRAW P.'40+X,Y%(A.P.'50+X,Y%(T))
770  ?&600D=?&6002 4,N.T
780  F.T=0TO 50 WAIT N.T CLEAR0 P.$30''
790  P.' PAR VOTRE FAUTE LA BASE EST"
800  P."DETRUITE..."'  F.Z=0TO60 WAIT N. P.'PERSONNE NE PEUT"
810  P." VOUS PORTER   "'SECOURS...'' F.Z=0TO60 WAIT N. P. '
820  P." VOUS ATTENDRE DONC LA MORT "'DANS DE TERRIBLES '
830  P."SOUFFRANCES   ''P.'          " F.Z=0TO20 P.'+'N. G.1
840 REM agrandissement de l'image
850 CLEAR0 F.Z=0TO64 MOVE X DRAW Y N.
860  U=1 X=X-32 MOVE25 Y%(25)-1 PLOTT 37 Y%(25)-1 P.
870 F.Z=0TO30 WAIT N. RU
```

Description du programme

Ligne 110 (initialisation Y) : ordonnée de la position du module ; (initialisation y) : vitesse aléatoire horizontale du module ; (initialisation U) : agrandissement réalisé ou pas ; (initialisation V) : vitesse verticale nulle.
Ligne 140 : initialisation y (x), x ∉ [0,130], ordonnée du sol lunaire.
Ligne 160 : y (o) ordonnée du sol d'abscisse 0 ou départ aléatoire ±20.
Lignes 170 à 200 : calcul des autres ordonnées par rapport aux premières pour qu'il y ait continuité (écart ±2).
Lignes 220 à 270 : idem.
Ligne 290 : traçage sur l'écran du sol lunaire (x = 64 abscisse de départ du module)
Lignes 300-310-320 : « fignolage »
Ligne 350 : extinction de l'ancienne position du module.
Lignes 370 à 390 : lecture des touches.
Ligne 410 : calcul de la nouvelle position.
Lignes 420-430 : contrôle de sortie de l'écran.
Ligne 460 : temporisation (si le programme est trop rapide on s'écrase).
Lignes 480-490 : on vérifie si le module touche le sol, auquel cas on examine sa vitesse.
Ligne 510 : si le module n'est pas assez près de la base on agrandit l'image (V).
Ligne 520 : retour en 350.
Lignes 550-560-570 : si le module a aluni en dehors de la base, des commentaires s'imposent.
Lignes 580 à 640 : la vitesse est supérieure à 4, on dessine l'explosion et on fait des commentaires.
Lignes 660 à 700 : dessin du module.
Lignes 720 à 830 : commentaire ou explosion.
Ligne 850 : agrandissement de l'image en passant à une définition inférieure de l'écran.
Ligne 870 : attend environ 5 s puis renvoie au début du programme.

panique au fond des mers

Vous voici aux commandes d'un sous-marin possédant un tube lance-torpilles. Votre but ? Abattre le plus d'adversaires possibles. Rassurez-vous : à défaut de vous transporter dans un espace intergalactique, vous traverserez les marécages de la haute résolution du Vic-20.

Premiers parution : L'Ordi n° 46

Tapi au fond de la mer, vous êtes en possession d'une arme vous permettant de détruire trois types d'ennemis : un hélicoptère (survolant la mer, bien sûr !) qui vous rapporte 100 points, deux bateaux, pour lesquels vous bénéficierez de 70 points, un sous-marin grâce auquel vous gagnerez 200 points.

A chaque passage le sous-marin dépose une mine, dans le seul but de vous faire perdre 15 points, mais vous obtenez alors une torpille de plus (vous disposez initialement de soixante-six torpilles). Si vous touchez un sous-marin, toutes les mines déjà présentes à l'écran disparaîtront.

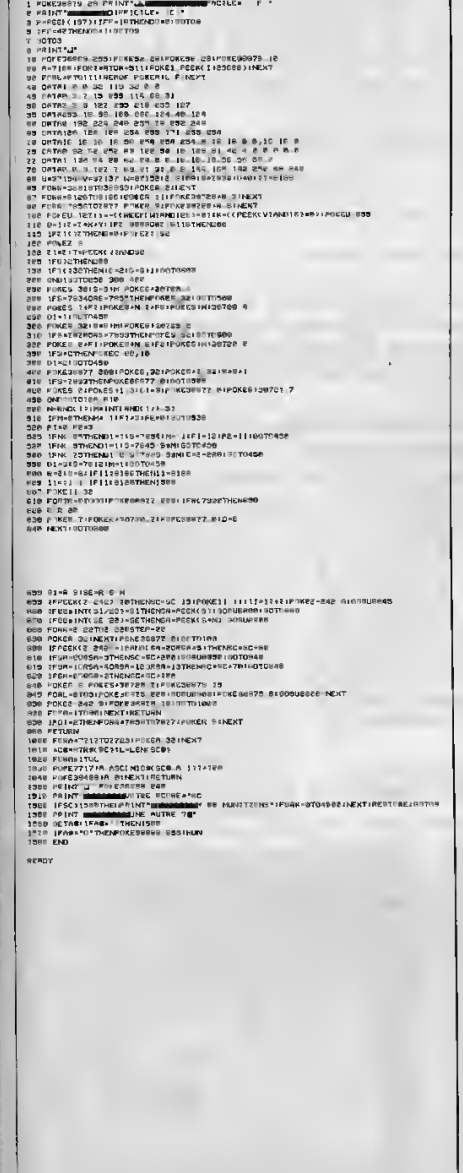
Pour éviter le tir en rafales, un test vérifie que vous n'avez pas tiré trois coups successifs du même endroit, auquel cas le canon deviendrait inactif (vous devrez alors le déplacer pour pouvoir l'utiliser normalement).

La mise à feu de la dernière torpille provoque l'affichage de votre score définitif, ainsi que la fin du jeu si vous n'avez pu dépasser les 1500 points nécessaires à une nouvelle partie.

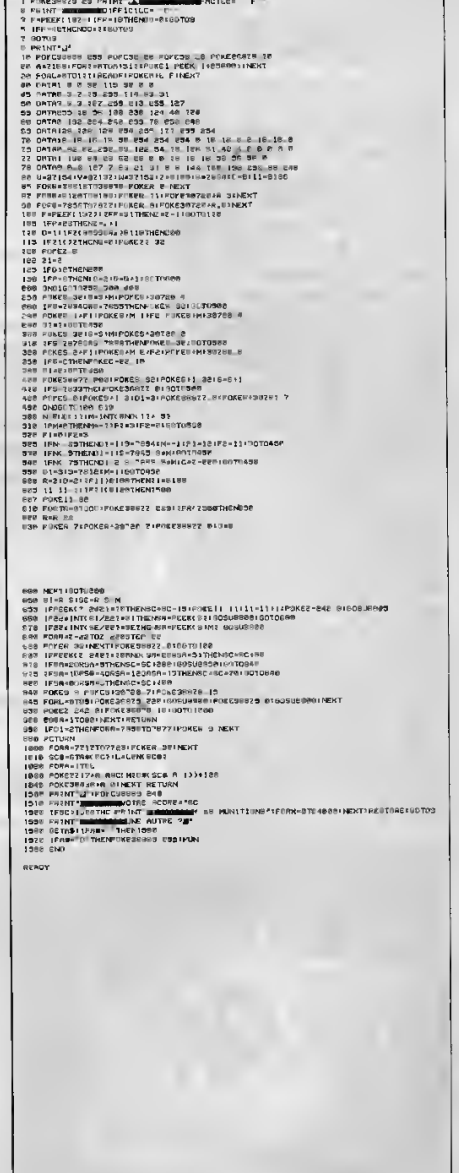
Un programme avec ou sans poignée de jeu

Le programme, tel qu'il est écrit, nécessite l'emploi d'un manche à balai (CBM ou Atari), mais le tableau 1 *(page 86)*, vous permettra d'utiliser le clavier sans pour autant altérer l'intérêt du jeu.

Programma avec poignée de jeu



Programma sans poignée de jeu



D'autre part, afin de ne pas dépasser la capacité de mémoire vive du système (3 583 octets libres), il n'y a pas de REM coûteux (mais vous pouvez consulter le tableau 1 concernant la description du programme).

Peut-être vous étonnerez-vous de la taille imposante de ce programme pour un tel jeu. Il aurait en effet été plus économique d'insérer une routine d'affichage et de déplacement des caractères dans une boucle FOR NEXT et de s'en servir successivement pour chaque adversaire (quatre ennemis, le canon, la rafale). Mais cette méthode présente le grand désavantage de nécessiter une batterie impressionnante de tests du type IF AND THEN ou IF OR THEN, ce qui ralentit l'exécution ; d'où l'obligation d'écrire une routine propre à chaque objet.

Si vous désirez modifier le graphisme, vous pouvez recalculer les argumants des données (lignes 40 à 79), suivant les indications de l'encadré 2.

Le processeur 6502 « comprend » un caractère comme une suite de huits nombres X (o<X<255), chaque nombre X étant la représentation décimale d'un octet. Voici donc la procédure à suivre pour obtenir un nouveau caractère :

. tracez sur une feuille une matrice de 8×8 cases ;

. exécutez le dessin de votre choix dans cette matrice, en sachant qu'une case ne peut être que vide ou pleine (0 ou 1) ;

. pour chaque ligne, faites la somme des huits cases qui la composent : une case i prend la valeur 0 si cette case est vide ou la valeur 2^i si cette case est pleine ($0 \leq i \leq 7$) ; la ligne 0 est la ligne du haut, la colonne 2^0 est la colonne la plus à droite ;

. reportez les huit arguments dans le DATA concerné, par exemple :

$2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

Ligne 1 [■ ■ ■ ■ ■ ■ ■ ■] = 1 octet

La valeur de cette ligne sera donc : Li = 0 + 64 + 32 + 0 + 0 + 4 + 2 + 0 = 102

A vous de jouer maintenant. Si vous disposez d'une extension mémoire, vous pouvez même ajouter diverses fioritures comme un graphisme plus réaliste ou des adversaires plus coriaces (poulpes géants ou cétacés féroces).

Patrick Lenoble

Tableau 1 : quelques petits conseils pour modifier le programme, de façon à pouvoir utiliser le clavier pour contrôler les déplacements et le tir.

Ligne	Commentaires
80	effacer : U=37154 V=37137 W=37152 uniquement
100	effacer complètement et remplacer par la ligne ci-dessous
100	écrire P=PEEK(197): IFP=31 THEN Z=Z-1.GOTO110
105	écrire IF P = 23 THEN Z=Z+1
110	effacer seulement Z=Z+X+Y:
122	effacer uniquement T=PEEK(V)AND32
130	remplacer T<>32 par P=8

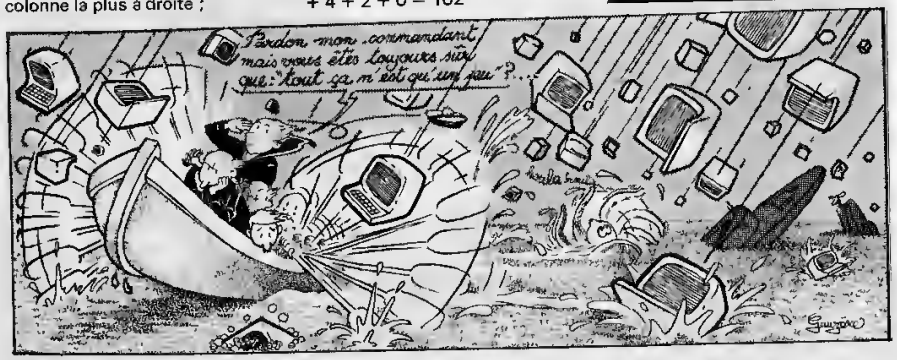
Touches à utiliser

Touche	Commentaire	Code
↑ CRSR	déplacement à gauche	31
↓ ←CRSR→	déplacement à droite	23
←	pour tirer	8

Bien entendu, pour changer de touches, il vous suffira de remplacer les codes 31, 23 et 8 des lignes 100, 105 et 130 par ceux de votre choix.

Tableau 2 : calcul des arguments des DATA permettant de modifier les graphismes.

Ligne	Caractère au clavier ou POKE----, X	Commentaires
40	, X=0	arrière de l'hélicoptère
45	A , X=1	arrière du bateau n° 1
50	B , X=2	arrière du sous-marin
55	C , X=3	avant de l'hélicoptère
60	D , X=4	avant du bateau n° 1
65	E , X=5	avant du sous-marin
70	F, G , X=6, X=7	canon, rafale
75	H, I , X=8, X=9	explosion, vague
77	J , X=10	mine
79	K, L , X=11, X=12	arrière et avant du bateau n° 2



autocritique aléatoire

votre ordinateur avoue que ses dés sont pipés, mais vous propose en échange un programme d'histogramme

Dans de nombreuses applications, il est intéressant de connaître la répartition d'une série de valeurs numériques : analyse de sondages, résultats expérimentaux, etc. Pour cela, on peut visualiser l'ensemble de ces valeurs sur un dessin à deux dimensions, en tenant compte de la fréquence d'occurence de chacune d'elles. La figure ainsi obtenue s'appelle un *histogramme* (du grec Histos, « texture, trame »).

Son aspect est un peu celui de nos cités modernes. On le retrouve dans la plupart des ouvrages techniques contemporains. Si la collecte des valeurs peut être réalisée de manières très diverses, leur exploitation (tracé de l'histogramme) est un travail facile à effectuer avec un ordinateur individuel. Dans ce domaine encore, une telle machine, posée sur le bureau de l'homme d'affaires, de l'ingénieur ou de l'étudiant, peut rendre de grands services. Encore faut-il l'avoir correctement programmée.

Le programme que nous vous présentons vous permettra de réaliser des histogrammes de manière assez générale.

Il est écrit ici pour un TRS-80, mais peut être transformé pour fonctionner sur d'autres ordinateurs individuels. Notre programme est écrit en BASIC. Les données à analyser peuvent figurer dans le corps du programme sous forme de DATA ou être calculées par ordinateur.

Pour chaque valeur, il s'agit de visualiser sur l'écran un « pavé » dont la surface corresponde à la fréquence de rencontre de cette valeur. Pour cela, on rencontre deux principales difficultés.

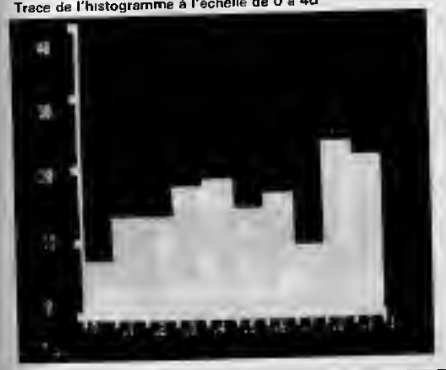
La première réside dans le fait que l'intervalle des valeurs peut être quelconque. Il importe que cet intervalle apparaisse en entier sur l'écran. Il faut donc, lors de la prise en compte des valeurs, repérer la plus petite et la plus grande et fixer aussi l'échelle de l'histogramme.

Nous ferons figurer l'intervalle en abscisse en le divisant arbitrairement en un certain nombre de parties (dix par exemple).

La seconde tient au nombre quelconque de valeurs. Le nombre est visualisé sur l'axe des Y.

Pour résoudre le problème, notre programme peut graduer de deux manières l'axe des ordonnées. Lorsque chaque pavé contient au plus 40 valeurs, la graduation va de 0 à 40. Quand cette limite est dépassée, l'axe des ordonnées est gradué suivant une échelle logarithmique : au lieu de représenter des longueurs proportionnelles à une valeur, on représente des longueurs

Trace de l'histogramme à l'échelle de 0 à 40



Tracé de l'histogramme avec l'échelle logarithmique

Première parution : L'Ordi n° 14

proportionnelles à son logarithme. (*)

Dans ce cas l'axe des y comporte cinq divisions qui correspondant à 1, 10, 100, 1 000, 10 000 valeurs Après la tracé des axes et de leurs graduations, nous construirons l'histogramme

Pour chaque valeur, nous rechercharons sa position dans un des crénaux déterminés précédemment et ajouterons d'une unité la hauteur du «pavé» correspondant Avant l'affichage, la hauteur de chaque pavé sera éventuellement corrigée dans le cas de l'emploi de l'échelle logarithmique

Exemple d'utilisation

Une utilisation intéressante est la suivante. Notre ordinateur individuel pourvu d'une fonction RND (0) qui permet de tirer des nombres au hasard compris entre 0 ,000001 et 0 ,999999 inclus Nous souhaiterions vérifier dans quelle mesure ce «hasard» correspond bien au vrai hasard

Dans ce but nous allons faire fonctionner la loterie BASIC correction écran un grand nombre de fois (10 000 par exemple) A chaque tirage, notre programme recherchera dans quelle région de l'intervalle 0-1 se situe le nombre tiré Une fois les tirages terminés, il en tracera l'histogramme L'examen de celui-ci nous permettra peut-être de déceler une «préférence» du générateur de nombres aléatoires, «auquel cas cas nombres ne sont peut-être pas tout à fait aléatoires»

Pour cette application, nous devrons modifier quelque peu notre programme La série des valeurs est obtenue par la fonction RND (0) au lieu d'être lue an DATA par des READ

Structure du programme

Lignes 40-45. réservation des tableaux nécessaires au programme tableau contenant les valeurs à analyser et tableau contenant le nombre de valeurs de chacun des dix créneaux

Ligne 50: appel du sous-programme d'acquisition des valeurs Le sous-programme peut, comme ici, lire des valeurs en DATA ou bien les calculer (par exemple à l'aide de la fonction RND)

(*) *Avec une échelle normale, représenter 10, 100, 1 000, se fera par exemple avec des longueurs 10, 100, 1 000 Avec une échelle logarithmique, ce serait par exemple 20, 30 40 On économise donc de la place*

Liste du programme

```
10 REM *** REALISATION D'HISTOGRAMMES ***
20 REM COPYRIGHT L'ORDINATEUR INDIVIDUEL
30 POKE 16553,255:REM *** SPECIAL TRS-80 ***
40 DIM V(500):REM POUR LES VALEURS
45 DIM Z(10) :REM POUR LES CRENEAUX
50 GOSUB 1000                    :REM ACQUISITION DES DONNEES
60 IN=(MA-MI)/10
65 Z(10)=1:PRINT@530, "*** CLASSEMENT DES VALEURS ***"
70 FOR N=1 TO NH                 :REM REMPLISSAGE DES TABLEAUX
80    FOR J=1 TO 10
90       IF V(N) >= (MI+(J-1)*IN) AND V(N) < (MI+J*IN)
         THEN Z(J)=Z(J)+1:J=10
110   NEXT J
120 NEXT N
125 EL=0                         :REM ECHELLE LOGARITHMIQUE NECESSAIRE ?
130 FOR J=1 TO 10
135    IF Z(J) > 40
       THEN EL=1:GOSUB 3000:J=10  :REM REDUCTION POUR ECHELLE LOG
140 NEXT J
145 REM ----------------------------------------------------------
150 REM  TRACE DES AXES ET DES GRADUATIONS
155 CLS
160 IF EL=0 THEN K=40 ELSE K=10000
165 FOR J=64 TO 896 STEP 192     :REM VALEURS SUR AXE DES Y
170    PRINT@J,K
180    IF EL=0 THEN K=K-10 ELSE K=K/10
190 NEXT J
200 FOR J=1 TO 41 STEP 10        :REM GRADUATION AXE DES Y
210    SET(13,J)
220    SET(12,J)
230 NEXT J
240 FOR Y=1  TO 41               :REM TRACE AXE DES Y
250    SET(14,Y)
260 NEXT Y
270 FOR X=14 TO 114
280    SET(X,41)
290 NEXT X
300 K=903
310 FOR J=1 TO 11                :REM VALEURS SUR AXES DES X
320    PRINT@K, MI+(J-1)*IN
340    K=K+5
350 NEXT J
360 FOR X=14 TO 114 STEP 10      :REM GRADUATIONS AXE DES X
370    SET(X,42)
380 NEXT X
390 REM ----------------------------------------------------------
400 REM TRACE DES PAVES DE L'HISTOGRAMME
410 FOR J=1 TO 10
425    FOR Y=41 TO 41-Z(J) STEP -1
430       FOR X=14+(J-1)*10 TO 14+J*10-1
440          SET(X,Y)
450       NEXT X
460    NEXT Y
470 NEXT J
480 INPUT RS
490 CLS
500 END
510 REM ----------------------------------------------------------
1000 REM SOUS-PROGRAMME D'ACQUISITION DES VALEURS
1005 REM CAS DE VALEURS EN DATA
1010 CLS
1020 INPUT"NOMBRE DE VALEURS A ANALYSER ";NH
1030 CLS
1040 RESTORE
1050 READ V(1)
1060 MI=V(1):MA=MI
1070 FOR N=2 TO NH
1080    READ V(N)                :REM DETERMINATION MINI ET MAXI
1090    IF V(N) < MI THEN MI=V(N)
                 ELSE IF V(N) > MA THEN MA=V(N)
1100 NEXT N
1110 RETURN
1500 REM ----------------------------------------------------------
3000 REM REDUCTION POUR ECHELLE LOG
3005 Z9=LOG(10)                  :REM CONVERSION LOG NEPERIEN
3010 FOR K=1 TO 10               :REM EN LOG DECIMAL
3020    IF Z(K) <> 0 THEN Z(K)=INT(LOG(Z(K))*10/Z9)
3030 NEXT K
3040 RETURN
5000 REM ----------------------------------------------------------
5010 REM VALEURS EN DATA
5020 DATA 10,11,5,6,7,11,28,56,57,59,34.5,35,89,87,86,88,85,76
5030 DATA 45,45,89,90,95,95,95,95,43,78,78,56,99,22,21
5040 DATA 0,0.5,2,2,2,2,2,2,2,100,90
5050 DATA 45,45,45,45,45,45,46,47,37,48,48,48,49
```


modifications à apporter à notre programme pour test du générateur de nombres aléatoires

```
50 MI=0: MA=1                    :REM *** VALEURS GENEREES PAR RANDO( ***
60 IN=(MA-MI)/10
65 Z(10)=1
70 CLS:INPUT"NOMBRE DE TIRAGES AU SORT ";N
75 FOR N=1 TO N: PRINT@534, N    :REM TIRAGE AU SORT
80    Z=RND(0)
90    FOR J=1 TO 10
95       IF Z >= (MI+(J-1)*IN) AND Z < (MI+J*IN)
         THEN Z(J)=Z(J)+1:J=10
110   NEXT J
120 NEXT N
```

Ligne 60 calcul de l'intervalle entre deux divisions de l'axe des X

Ligne 65 cette ligne est nécessaire pour tenir compte de la borne supérieure de l'intervalle des valeurs la valeur V(N) correspondant à MA = MI + 10*IN n'est en effet pas prise en compte à la ligne 90, et la ligne 65 y remédie

Ligne 70-120 pour chaque valeur, recherche de son emplacement dans un des dix creneaux

Lignes 125-140 recherche d'un créneau contenant plus de 40 valeurs Si l'on en trouve un, on passe en échelle logarithmique sur l'axe des y (indicateur EL) La reduction est effectuée par l'appel du sous-programme situé en lignes 3 000 3 040

Ligne 150-380 tracé des axes x et y, de leurs graduations et des valeurs correspondantes Sur l'axe des y, deux types de graduations sont possibles (arithmétique et logarithmique)

Lignes 400-470 pour chaque créneau, tracé d'un pavé dont la hauteur correspond au nombre de valeurs contenues dans ce creneau Cette hauteur peut être éventuellement corrigée pour tenir compte de l'échelle logarithmique

Lignes 1 000 1 110 acquisition des valeurs Les valeurs sont ici données en DAT On determine les bornes inférieure et supérieure de l'intervalle des valeurs

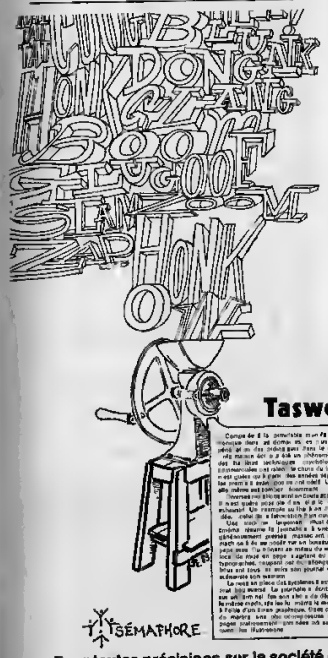
Lignes 3 000-3 040 réduction de la hauteur de chaque pavé dans le cas de l'utilisation d une échelle logarithmique

Lignes 5 010-5 050 valeurs en DATA

Notre programme utilise certaines particularites du BASIC TRS-80 PRINT @ 100 écrit à partir du 100e caractere de l'écran, qui en comporte 16 × 64 = 1 024 SET (X,Y) allume sur l'écran un pavé situé à la ligne Y de la colonne X, X est compris entre 0 et 127, et Y entre 0 et 47 Ces instructions PRINT @ et SET devront être remplacées par des PRINT simples ou des POKE judicieux pour que le programme fonctionne sur d'autres machines

Nous reproduisons deux histogrammes obtenus successivement et portant chacun sur 10 000 tirages Seule la comparaison entre eux d'un grand nombre d'histogrammes pourrait nous renseigner valablement sur le bon fonctionnement de notre loterie

Daniel Lucet



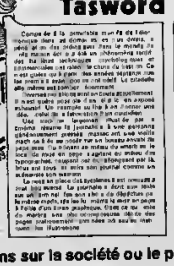
TASWORD DEUX

LE TRAITEMENT DE TEXTE PROFESSIONNEL POUR SPECTRUM 48K.

Adaptation française de Tasword Two maintenant disponible vous offre :
- *caractères accentués, cédilles, thèmes etc.*
- *64 caractères par ligne sur l'écran et l'imprimante (idéal pour A4).*
- *rapidité (écrit en code machine) mais personnalisable,*
- *toutes les fonctions habituelles justification commutable, insertions, déplacement de lignes et paragraphes, formattage et margination, remplacement de mots ou groupes de mots, etc...*
- *manuel complet et instructions d'écran en français,*
(manuel aussi livrable en allemand et néerlandais)
- *programme d'apprentissage et d'entraînement Tasword Tuteur,*
- *utilise sans interface l'imprimante ZX,*
- *compatible avec la majorité des interfaces et imprimantes professionnelles (Cobra RS 232 i/o port, Euroelectronics ZX Print, Kempston, Hilderbay, Morex, Tasman, etc... - Epson, Seikosha, Oki, Tandy, etc...)*

Nous recommandons et fournissons l'interface Tasman centronic, programmée par Tasword Deux pour l'Epson FX-80 et livrée avec programmes sur cassette pour les Star et Tandy graphique couleur.
Une notice détaillée permet l'adaptation par l'utilisateur.

Tasword Deux 60f.s. Interface 150f.s. Les 2: 200f.s.



SEMAPHORE

VENTE PAR CORRESPONDANCE, *envoyer à* SEMAPHORE *Logiciels,*
Jean-Pierre Cardinaux
CH 1283 La Plaine
(Genève) Suisse.

Je commande : *Tasword deux*
Tasman interface
Je verse le montant de F*s, au CCP (ou chèque) 12.247 98*
de Jean-Pierre Cardinaux à Genève.
Je verse au compte CO-192.930.1 à la Société de Banque Suisse à Genève.
Contre remboursement (taxes en plus)
Nom: *prénom*
adresse :
Téléphone :

Pour toutes précisions sur la société ou le produit présenté ci-dessus : Référence 110 du service-lecteure (psge 172)

tracé de courbes

arabesques et entrelacs

Votre ordinateur ou votre calculatrice scientifique peut évaluer de très nombreuses fonctions mathématiques. Si elles ne sont pas toutes préprogrammées, il ne tient qu'à vous de les obtenir au moyen d'un petit sous-programme. Les ordinateurs excellent dans l'art de traiter les nombres, mais ils ont aussi des fonctions graphiques. Autant dire qu'ils sont tout indiqués pour tracer automatiquement les courbes représentant les fonctions. On peut, à volonté, leur demander de faire serpenter celle d'un sinus ou de visualiser la croissance exponentielle d'une population. Ne nous en privons pas.

Si vous avez accès à un TRS 80 modèle 1, 16 Ko, il vous suffit d'introduire ce programme BASIC pour obtenir la représentation graphique de toute fonction f(x). Bien entendu, nous vous faisons confiance pour adapter ce programme à votre ordinateur favori

Sans crayon sans gomme et... sans papier

Vous vous souvenez sans doute d'énoncés du genre « soit la droite d'équation y = x-4 » ou « soit la parabole y = -x² + 5 ». Si vous avez une grande habitude des fonctions, cela vous suffit, on peut même dire que vous « voyez » la droite ou la courbe. Il en va de même pour le solfège : les virtuoses de la lecture musicale lisent la partition et l'entendent ! Mais ceux qui ne sont pas rompus à ce type d'exercices n'y entendent rien...

Si vous êtes comme eux, vous ne savez pas, « à vue de nez », de quel côté la parabole est orientée, vous ne savez pas si la droite est croissante ou décroissante. Pour en avoir une idée, vous devez calculer quelques points de chaque fonction et les représenter sur une feuille de papier.

Pourquoi votre ordinateur ne le ferait-il pas sur son écran ? Il vous suffira de définir les deux fonctions conformément au cadre du programme et de taper RUN.

L'ordinateur vous demandera quelques précisions, telles que les valeurs extrêmes de x et de f(x) à représenter. La raison en est simple : les dimensions de l'écran comme celles de votre feuille de papier, sont finies. Autant indiquer tout de suite la portion de courbe qui vous intéresse. Il est en effet inutile de représenter la fonction pour x = 100 si vous vous intéressez seulement aux valeurs proches de zéro. Pour les deux fonctions que nous avons retenues, explorez par exemple le domaine allant de −10 à +10 (tant pour x que pour f [x]).

Vous verrez d'abord apparaître une croix appelée système d'axes de coordonnées qui sert de repère. La droite qui apparaît ensuite représente la première fonction (y = x − 4) et vous verrez ensuite se dessiner une parabole définie par y = x² + 5.

Le domaine de variation est indiqué dans le coin inférieur gauche. La variable x est représentée sur l'axe horizontal et y (= f [x]) sur l'axe vertical. La graduation est la

Principales variables utilisées

N	Nombre de fonctions
x1, x2, y1, y2	Domaine de variation de x et y (= f [x])
Gx, Gy	Graduation des axes x et y
Ax, Ay	Coordonnées de l'intersection des axes
O$	Contient la réponse à la question « arrêt ou non après chaque fonction ? »
K, R	Sont des compteurs

Première parution : L'OI n° 31


```
10 REM  * REPRESENTATION GRAPHIQUE DE FONCTIONS A UNE VARIABLE *
20 REM     MIS AU POINT SUR TRS-80
30 REM  AUTEUR : JEAN-PAUL ZANTER
40 REM  COPYRIGHT L'ORDINAT. INDIVIDUEL ET L'AUTEUR
50 REM  ***********************************************
100 REM          DEBUT DU PROGRAMME
110 CLS
120 PRINT @94,"REPRESENTATION GRAPHIQUE"
130 PRINT @145,"DE FONCTIONS A UNE VARIABLE"
140 PRINT @320,"AVEZ-VOUS INTRODUIT LA BOUCLE"
150 PRINT @384,"N" : REM TEST SUR LA REPONSE
160 IF N$="N" THEN GOSUB 1000 : REM MODE D'EMPLOI
170 REM          INITIALISATIONS
180 CLS
190 PRINT@576,"COMBIEN DE FONCTIONS AVEZ-VOUS INTRODUITES?"
200 INPUT N
210 IF N<=0 OR N>5 THEN 200
220 IF N=1 THEN 270
230 PRINT@704,"VOULEZ-VOUS UN ARRET APRES CHAQUE FONCTION?"
240 GOSUB 1200 : REM CHOIX D'UNE GRADUATION
250 GOSUB 9000
260 GOSUB 1200
270 REM          REPRESENTATION DES FONCTIONS
280 REM
290 FF=0
300 FOR K=1 TO N
310      FOR R=1 TO 113 : REM CALCUL POINT PAR POINT
320           GOSUB 1500
330 NEXT R
340 GOSUB 1700 : REM GRADUATION
350 IF O$="N" THEN 400 : REM PAS D'ARRET
360 IF K=N THEN 400
370 IF FF=1 THEN 400 : REM AUTRE GRADUATION?
380 IF O$="O" THEN FF=1: K=N: GOTO 400
390 GOSUB 2000 : REM BOUCLE D'ATTENTE
400 NEXT K
410 IF FF=1 THEN 270 : REM AUTRE GRADUATION?
420 GOSUB 1900
430 PRINT @704,"ARRET DU PROGRAMME";
440 PRINT @1000,"PAR <BREAK>";
450 R$=INKEY$
460 IF R$<>"BREAK" THEN 450
470 END
1000 REM*****************************************************
1010 REM          MODE D'EMPLOI DU PROGRAMME
1020 CLS
1030 PRINT"VOUS POUVEZ REPRESENTER JUSQU'A 5 FONCTIONS"
1040 PRINT"INTRODUIRE LA PREMIERE FONCTION A LA LIGNE 2120"
1050 PRINT"PUIS LA DEUXIEME FONCTION A LA LIGNE 2220"
1060 PRINT"PUIS AUX LIGNES 2320, 2420 ET 2520"
1070 PRINT"COMME DANS L'EXEMPLE SUIVANT:"
1080 PRINT"PRINT" 2120 F(1)=SIN(X)"
1090 PRINT" 2220 F(2)=COS(X)"
1100 PRINT"PRINT" LA VARIABLE DOIT ETRE X"
1110 PRINT"OBLIGATOIRE
1120 PRINT:PRINT"CECI FAIT, TAPEZ ' RUN ' SUIVI DE <ENTER>"
1130 PRINT
1140 STOP
1150 RETURN
1200 REM*****************************************************
1210 REM          CHOIX DES GRADUATIONS
1220 CLS
1230 PRINT @576,";
1240 INPUT "REPRESENTATION GRAPHIQUE POUR X ALLANT DE...A..";X1,X2
1250 IF X1>X2 THEN 1240
1260 INPUT "REPRESENTATION GRAPHIQUE POUR F(X) ALLANT DE...A...";Y1,Y2
1270 IF Y1>Y2 THEN MX=Y1: Y1=Y2: Y2=M
1280 IF Y1=Y2 THEN 1270
1300 GX=(X2-X1)/100 : REM GRADUATION DE L'AXE HORIZONTAL
1310 GY=(Y2-Y1)/40 : REM GRADUATION DE L'AXE VERTICAL
1320 IF X1>0 THEN AX=1: ELSE AX=X1/GX : REM AXES DE COORDONNEES
1330 IF Y1>0 THEN AY=1: ELSE AY=Y1/GY
1340 IF Y1*Y2 <= 0 THEN AY=Y2/GY : ELSE AY=113
1350 FOR X=1 TO 113
1360      SET (X,AY)
1370 NEXT X
1380 FOR Y=0 TO 59
1390      SET (AX,Y)
1400 NEXT Y
1410 RETURN
1500 REM*****************************************************
1510 REM          CALCUL POINT PAR POINT DES FONCTIONS
1520 Z=X*(R-1)*GX
1530 ON K GOSUB 2100,2200,2300,2400,2500
1540 IF F(K)>Y2 OR F(K)<Y1 THEN 1590
1550 Y=(Y2-F(K))/GY
1560 SET (R,Y)
1570 RETURN
1580 SET (R,AY)
1590 F(K)=Y2+1
1600 RETURN : REM FONCTION NON DEFINIE
1700 REM*****************************************************
1710 REM          GRADUATION
1720 FOR X=AX-40 STEP 20
1730 FOR Y=2*14 AND X<=113 THEN RESET (X,AY)
1740 NEXT X
1750 FOR Y=AY-40 TO AY+40 STEP 5
1760 NEXT Y
1770 IF Y>=0 AND Y<=59 THEN RESET (AX,Y)
1780 PRINT @896,X1;"<X<";X2; TAB(13) " 1 GRADUATION =";GX*20;
1790 PRINT @960,Y1;"<Y<";Y2, TAB(13) " 1 GRADUATION =";GY*5;
1800 RETURN
1900 REM*****************************************************
1910 REM          CHOIX D'UNE AUTRE GRADUATION
1920 PRINT @934,"UNE AUTRE GRADUATION";
1930 PRINT @1000,"VOULEZ-VOUS UNE";
1940 GOSUB 9000 : REM AUTRE GRADUATION?
1950 RETURN
2000 REM*****************************************************
2010 REM          BOUCLE D'ATTENTE
2020 PRINT @934,"POUR CONTINUER,  ";
2030 PRINT @1000,"TAPER ?  ";
2040 R$=INKEY$
2050 IF R$<>"" THEN 2040
2060 PRINT @934,STRING$(17," ");
2070 PRINT @1000,STRING$(17," ");
2080 RETURN
2100 REM*****************************************************
2110 REM          PREMIERE FONCTION
2120 F(1)=X-4
2130 RETURN
2200 REM*****************************************************
2210 REM          DEUXIEME FONCTION
2220 F(2)=-X*X*5
2230 RETURN
2300 REM*****************************************************
2310 REM          TROISIEME FONCTION
2320 F(3)=SQR(X)
2330 RETURN
2400 REM*****************************************************
2410 REM          QUATRIEME FONCTION
2420 F(4)=SGN(X*X*X*5*X*5)
2430 RETURN
2500 REM*****************************************************
2510 REM          CINQUIEME FONCTION
2520 F(5)=0
2530 RETURN
9000 REM*****************************************************
9010 REM          TEST SUR LA REPONSE
9020 R$=INKEY$
9030 IF R$="" THEN 9020
9040 IF R$<>"N" AND R$<>"O" THEN 9020
9050 RETURN
```

distance entre deux points noirs sur l'un ou l'autre des axes de coordonnées. Ici, le point d'intersection des deux axes correspond à zéro.

Si vous désirez une autre graduation, rien de plus facile : vous pouvez agrandir une portion de l'écran afin de mieux voir l'une des intersections, par exemple celle située entre 0 et 4 pour x et 0 et −4 pour f(x). Notez bien que les valeurs positives sont toujours représentées à droite de l'axe des ordonnées et au-dessus de celui des abscisses.

Le graphique peut devenir esthétique

Il est normal, après cet effet de « zoom » que vous ne raconnaissiez plus la parabole : vous n'en avez qu'un détail.

En prenant une graduation de plus en plus fine, vous connaîtrez la position de l'intersection avec une précision de plus en plus grande. En fait, vous allez pouvoir résoudre graphiquement un système d'équations de la façon la plus commode possible. De la même façon, vous déterminerez graphiquement le zéro d'une fonction, c'est-à-dire la valeur de la variable x pour laquelle la fonction est égale à zéro.

Il arrive que la fonction ne soit pas toujours définie. Ainsi SQR (x) n'est définie que pour les valeurs positives de x. Si vous demandez la représentation graphique de x allant de −4 à +4, l'axe horizontal sera effacé jusqu'à ce que la fonction soit définie. Vous comprendrez aisément en faisant quelques essais.

Pour vous faire la main, regardez donc ABS (x), INT (x), SIN (x)/x, SIN (3 * x) * EXP (-x/5), SQR (x * x * x + 3* x * x), etc.

Naturellement, une fonction peut être définie par morceaux :

```
100 : IF x < - 5 THEN F(1) = - 2 : GOTO 150
110 : IF x = - 5 THEN F(1) = 2 : GOTO 150
120 : IF x > - 5 AND x < 2 THEN F(1) = x
                          ELSE F(1) = 2
```

La représentation graphique s'inscrit dans un carré de 80 * 40 correspondant aux positions horizontales 14 à 113 et aux positions verticales 0 à 39.

Les différentes fonctions à représenter sont appelées aux sous-programmes 2100, 2200, 2300, 2400 et 2500.

Vous pouvez maintenant résoudre graphiquement des systèmes de fonctions et il n'est pas impossible que le tracé des courbes animées vous conduise à en rechercher quelques-unes pour des raisons d'ordre esthétique.

Jean-Paul Zanter

®AUTOCAD

CAO ET DAO EN DEUX DIMENSIONS POUR MICRO-ORDINATEUR

Le "Traitement de Texte" du dessin

DOMAINES D'APPLICATION

- Architecture
- Mécanique
- Electronique
- Topographie
- Bâtiment
- Publicité
- Design
- Imprimerie

Logiciel de dessin assisté par ordinateur en deux dimensions. Le logiciel permet le tracé automatique à un pas et à une échelle donnée. Il possède un zoom de facteur 1 million, il permet l'entrée de données sur table à digitaliser et Touch jeu. Sortie sur table traçante ou fichier, etc.

(1) CAO CONCEPTION assistée par ordinateur
(2) DAO DESSIN assisté par ordinateur
* marque déposée d'Auto Desk INC

NOUS RECHERCHONS VINGT-CINQ DISTRIBUTEURS RÉGIONAUX EXCLUSIFS

ZIP® COMPUTER
34, rue Vivienne, 75002 PARIS - téléphone : 233 72.07
Télex : G VIVIEN 670355 F

Pour toutes précisions sur la société ou le produit présenté ci-dessus : Référence 111 du service-lecteurs (page 172)

mettez de l'ordre dans vos données

grâce à ce programme de tri par insertion (*)

Pourquoi ne pas créer dans vos colonnes une nouvelle rubrique « algorithmes de tri » ?

En effet, d'une part de nombreuses applications informatiques nécessitent un tri, et d'autre part je pense qu'il est très difficile de trouver des algorithmes de tri efficaces pour un P.S.I. à langage interprété et disposant généralement de peu de mémoire ; si on n'y prend garde, on arrive facilement à des temps d'exécution de quinze minutes pour cent articles à trier, ce qui est évidemment excessif.

Pour faire un début, je vous propose ci-joint le meilleur algorithme de tri que j'aie trouvé à ce jour. Temps d'exécution de l'ordre de une minute pour trier 100 chaînes de 50 caractères alphanumériques chacune sur un PET 2001 de Commodore. Je souhaite que votre revue suscite une confrontation d'idées qui permettrait certainement d'en trouver de meilleurs, sans qu'il soit nécessaire de descendre au niveau du langage machine.

(*) *le tri par insertion est celui que fait un joueur de cartes classant son jeu.*

Ce programme présente l'avantage de ne pas trier la table des données de base T$, mais de ne travailler que sur la table de « pointeurs » A% (on appelle « pointeur » une variable dont la valeur désigne, « pointe vers », un élément). Ceci permet une plus grande rapidité d'exécution dans des cas bien précis : utilisé comme sous-routine, il permet de trier des chaînes de caractères relativement longues (plus de 5 caractères), mais il est préférable d'utiliser une méthode directe (trier directement dans A$) si l'on veut trier des chaînes de caractères courtes ou des nombres (*). C'est d'ailleurs la version « directe », sans pointeur, qui se trouve présentée dans la fiche pratique n° 11.

(*) *En fait, il semblerait que pour au moins tous les BASIC Microsoft, il soit préférable d'utiliser cette méthode dès qu'il s'agit de chaînes de caractères : ces BASIC « tassent » les chaînes de caractères, ce qui fait que chaque opération de chaîne entraîne systématiquement un « tassement ». Travailler avec ce système de pointeurs minimise les transferts de chaînes, et par conséquent le temps passé à tasser.*

BS

Première parution : L'O n° 8

Liste du programme en BASIC

```
10 REM TRI D'UNE TABLE EN MEMOIRE CENTRALE
15 REM PAR INSERTION DICHOTOMIQUE
20 REM AUTEUR   PIERRE KELLER
30 REM COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
40 GOSUB 100.REM INITIALISATIONS
50 GOSUB 200.REM IMPRESSION INITIALE
60 PRINT"QUAND VOUS VOULEZ POUR LE TRI":
   GOSUB9100 PRINT"OK"
70 GOSUB 400:REM TRI
80 PRINT"VOILA... LA LISTE TRIEE EST"
90 GOSUB 500.REM IMPRESSION APRES LE TRI
95 END

100 REM INITIALISATIONS
110 READN9 REM NOMBRE DE MOTS DANS LA LISTE (MOINS UN)
120 DATA48
130 DIM T$(N9)
135 DIM A%(N9).REM TABLEAU DES INDICES
140 FOR J=0TON9:READT$(J):NEXTJ
145 REM LES N9+1 MOTS DE LA LISTE
150 DATA CLR,DATA,DIM,END,FOR,GOSUB
155 DATA GOTO,INPUT,INT,LET,LIST,NEW
160 DATA NEXT,PRINT,READ,REM,RETURN
165 DATA RUN,SGN,SIN,SQR,STEP,STOP,TAB
170 DATA THEN,AND,CHR$,LOAD,COS,SAVE
175 DATA EXP,LEFT$,LEN,LOG,MID$,OR,NOT
180 DATA PEEK,POKE,RIGHT$,SPC,STR$,TAN
185 DATA TO,RND,VAL,WAIT,IF,ABS
199 RETURN

200 REM AFFICHAGE DES MOTS DE LA LISTE
210 REM LA LISTE EST DANS T$(N9)
220 FORJ2=0TON9
230 A$=T$(J2)+"          " REM ON AJOUTE 8 BLANCS DERRIERE LE MOT
240 PRINT LEFT$(A$,8);:REM AVANT DE N'EN PRENDRE QUE LES 8 PREMIERS CARACTERES
250 NEXTJ2
260 PRINT
299 RETURN

400 REM TRI DE LA TABLE T$(N9) DANS L'ORDRE CROISSANT
        (METHODE PAR INSERTION DICHOTOMIQUE ET TABLE)
405 FOR J4=0 TO N9 A%(J4)=J4 NEXTJ4
407 Z%=0:U%=1:D%=2:
    REM CONSTANTES ENTIERES POUR ALLER PLUS VITE
410 IF N9=0 THEN 499
420 N1=N9-1:IF T$(N9)<T$(N1) THEN
    A%=A%(N9):A%(N9)=A%(N1):A%(N1)=A%
425 IF N9=1 THEN 499
430 FOR J4%=N9-2 TO Z% STEP -U%
435 A8%=A%(J4%) A$=T$(A8%):
    IF A$<=T$(A%(J4%+U%)) THEN 460
440 INF%=J4% SUP%=N9+1
445 K4%=(INF%+SUP%)/D%:IF A$<T$(A%(K4%)) THEN SUP%=K4%:GOTO 455
450 INF%=K4%
455 IF SUP%-INF%>U% THEN 445
470 K1%=INF%-U%.FOR K4%=J4% TO K1%.A%(K4%)=A%(K4%+U%):NEXT K4%
475 A%(INF%)=A8%
460 NEXT J4%
499 RETURN

500 REM LISTE DE LA TABLE T$ DANS L'ORDRE DONNE PAR
        LA TABLE A%
510 FOR J5=0 TO N9
520  A$=T$(A%(J5))+"        ":
     REM 8 BLANCS DERRIERE LE MOT
530  PRINT LEFT$(A$,8);:REM 8 PREMIERS CARACTERES
540 NEXT J5
599 RETURN


9100 REM SOUS-PROGRAMME D'ATTENTE * TRS 80 NIVEAU 2
9110 A$=INKEY$.IF A$="" THEN 9110
9120 RETURN
9100 REM SOUS-PROGRAMME D'ATTENTE * PET
9110 GET A$.IF A$="" THEN 9110
9120 RETURN
9100 REM SOUS-PROGRAMME D'ATTENTE GENERAL
9110 INPUTA$
9120 RETURN
```

Remarques sur le programme

Il faut faire deux sous-programmes d'impression de la liste, un qui l'imprima dans l'ordre initial (lignes 200 à 299), l'autre qui imprima la liste par l'intermédiaire des pointeurs (lignes 500-599). La liste est alors imprimée triée.

Le manual d'utilisation du PET recommande, pour gagner du temps d'exécution, d'employer des variables plutôt que des constantes, et des variables entières (nom suivi de %) plutôt que des variables normales : ce qui explique la prolifération de % dans les lignes 405 à 480, et l'utilisation de Z%, U%, D% à la place des constantes 0, 1 et 2. Notez de même qu'à la ligne 445, comme K4% est entier, la division est une division « entière » (c'est-à-dire que 6/2 donne 3 et 7/2 donne 3 également).

L'impression peut se faire par blocs de 8 caractères tout en donnant une bonne présentation aussi bien sur le PET (lignes de 40 = 5×8 caractères) que sur les autres systèmes (lignes de 64 = 8×8 ou 80 = 10×8 caractères).

L'ordre des mots en DATA a été changé à la ligne 185, afin d'éviter les cas particuliers que ne permettait pas de détecter le fait qu'on termine par VAL et WAIT, qui sont également les derniers dans l'ordre alphabétique.

Notez la ligne 135, nécessaire puisqu'on utilise le tableau A % pendant le sous-programme de tri.

Le lecteur pourra tester son propre sous-programme de tri en le mettant en 400 à 499, et l'utilisation du sous-programme d'attente (9100-9120) permet de donner un « top chrono » bien pratique pour mesurer les performances du programme (sur des 48 + 1 = 49 chaînes utilisées ici). Les versions de ce sous-programme pour la PET et le TRS-80 sont fournies. Les systèmes ne disposant pas d'une commande de type GET (PET) ou INKEY$ (TRS-80) pourront utiliser la troisième version.

Enfin, signalons qu'il existe de nombreux ouvrages sur les tris, dont nous ne citerons que deux :

en français : G. Louit, « Algorithmes de tri », Dunod, Paris 1971, 158 pages ;

en anglais : D E Knuth, « The Art of Computer Programming », volume 3, « Sorting and Searching », Addison-Wesley, Mass. 1973.

Pierre Keller

postage

Simple... ...mais efficace

imprimer des adresses sur étiquettes

Certaines données, provenant d'un traitement par ordinateur, doivent être transcrites, sans erreur ni perte de temps, en un seul ou plusieurs exemplaires, sur des enveloppes, des fichiers, des dossiers ou autres documents. La réalisation de ce travail, manuellement ou avec l'aide d'une machine à écrire, entraîne un gaspillage d'argent, une perte de temps et de gros risques d'erreurs. Et pourtant, toute entreprise, petite ou grande, est confrontée chaque jour à des problèmes d'organisation de ce genre. Des étiquettes autocollantes en formats standardisés, en rangées simples ou multiples, sont vendues dans le commerce et s'adaptent à la plupart des besoins. Un ordinateur individuel, équipé d'une imprimante et d'un BASIC étendu permettant le traitement des variables alphanumériques, fixera « quelque part » ces informations qu'il vient d'élaborer.

Chaque variable alphanumérique peut contenir habituellement jusqu'à 255 caractères.

Dans le BASIC Microsoft, l'instruction CLEAR n (n étant une constante ou une expression) réserve n octets de la mémoire centrale pour le stockage de l'ensemble des variables alphanumériques ; n doit être égal ou supérieur au nombre *maximum* de caractères à stocker pendant le travail.

Remarque : en réservant le nombre exact d'octets, le programme gère efficacement la mémoire centrale de l'ordinateur ; un programme n'utilisant pas d'alphanumériques devrait comprendre un CLEAR 0 afin de supprimer le « CLEAR 50 automatique » que le BASIC réserve dès sa mise en route.

Le programme présenté ici utilise les fonctions suivantes :

LEFT$ (variable, n) qui donne les n caractères de gauche d'une séquence de caractères (left-gauche) ;

RIGHT$ (variable, n) qui donne les n caractères de droite d'une séquence de caractères (right-droite) ;

MID$ (variable, p, n,) qui donne les n caractères d'une séquence de caractères dont le premier est le (p) ième.

Un exemple d'application du programme

Le programme suivant :

```
10 A$ = « ADMINISTRATION »
20 PRINT LEFT$ (A$, 3)
30 PRINT MID$ (A$, 3,4)
40 PRINT RIGHT$ (A$, 6)
```

exécuté, donnerait :

```
ADM
MINI
RATION
```

L'instruction *LPRINT TAB (n) variables* déplace le curseur ou la

Première parution L'OI n° 9

tête imprimante vers la position désirée (n) de la ligne, avant d'écrire les différentes variables ; (n) ne peut dépasser le nombre maximum de caractères qu'une ligne peut contenir, par exemple 64 caractères pour écran vidéo et 132 caractères pour une imprimante. L'instruction TAB peut être utilisée plusieurs fois après LPRINT. Si l'on dépasse le nombre maximum de caractères d'une ligne, l'impression s'effectuera sur les lignes suivantes, mais la bel alignement risque de se trouver compromis !

Il faut définir des zones d'information

Supposons que l'on dispose de l'information à traiter en mémoire centrale, pour cela, la mémoire centrale doit être vaste ou les informations relativement peu nombreuses.

La traitement d'un DATA toujours le même nous oblige à définir les *zones d'information* (ce qui s'appelle « dassin de carte » dans le traitement par cartes perforées) Par exemple 25 caractères pour le nom, 19 caractères pour la rue et pour la ville, comme sur la représentation ci-contre

Chaque séquence devra respecter la disposition indiquée en *(1)*

(1)

```
10 REM PROGRAMME D'IMPRESSION D'ETIQUETTES
20 REM AUTEUR DANIEL WUESTENBERGHS
30 REM COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
40 CLEAR 1200
50 DATA "..............63 cars....................."
100 FOR X=1 TO N
110  READ A$, B$, C$
120  LPRINT          LEFT$(A$,25);:
     LPRINT TAB(26)  LEFT$(B$,25);:
     LPRINT TAB(52)  LEFT$(C$,25)
130  LPRINT          "RUE";MID$(A$,26,19);:
     LPRINT TAB(26)  "RUE";MID$(B$,26,19);:
     LPRINT TAB(52)  "RUE";MID$(C$,26,19)
140  LPRINT          RIGHT$(A$,19);:
     LPRINT TAB(26)  RIGHT$(B$,19);:
     LPRINT TAB(52)  RIGHT$(C$,19)
150  LPRINT" ";CHR$(10)
160 NEXT X
170 END
```

(2) Exemple de sortie d'étiquette

```
JEAN FRANCOIS                          ROBERT JACQUES
RUE DES PRES FLEURIS 2                 RUE DE LA FORE
1234 VILLE                             9876 VILLAGE
```

```
NNNN
```

urgent

aff. exception. : données à saisir

par prog. spécialisé et prat.

Lors de la création d'un fichier de données, que ce soit sur un gros ou un petit ordinateur, il faut trouver un moyen d'introduire dans le machine les différents éléments constituent ce fichier, et cele le plus commodément possible. Dans le cas des ordinateurs individuels, cette saisie s'effectue en « temps réel » c'est-à-dire directement au clavier, les renseignements étant immédiatement pris en compte. Cela peut présenter un certain danger : il convient de ne pes enregistrer des données erronées ! Une méthode de saisie doit essurer une certaine sécurité sur ce point en permettant à l'opérsteur (qui n'est que rarement informaticien) de toujours pouvoir rattreper ses erreurs. La facilité et le rapidité de travail sont également des critères importents. Nous vous proposons un systême de saisie qui tient compte de ces divers facteurs.

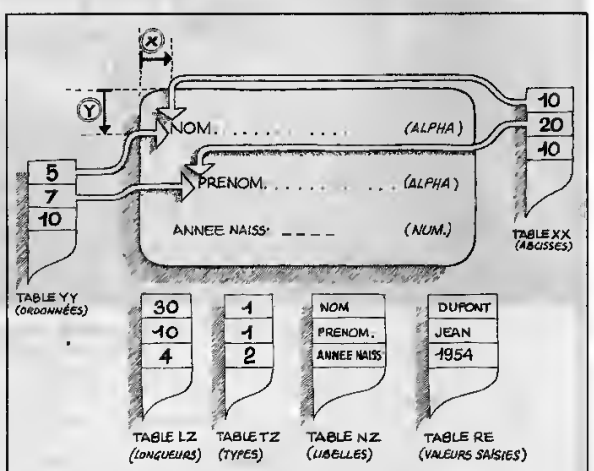
Pour chaque zone d'un enregistrement, le système doit interroger l'opérateur pour connaître la valeur à affecter à cette zone. Pour ce faire, lui et l'opérateur utilisent un langage commun de désignation dont les mots sont les libellés des zones. Il y aura autant de libellés que de zones différentes dans l'enregistrement. Il est très pratique de visualiser sur une même image d'écran l'ensemble de ces libellés, chacun etant suivi d'un certain nombre d espaces que l'opérateur remplira à partir du clavier. Cet imprimé ou «masque» sera le reflet exact de la structure du fichier. La disposition des libellés dans le masque relève d'un critère de lisibilite. D ou l'idée de faire correspondre à un fichier donné une table contenant les libellés des zones (Table NZ$). Une autre table donnera, pour chacune des zones, la position du debut de son libellé à l ecran (coordonnées X Y stockées dans les tables XX et YY par exemple)

Dans le fichier, chaque zone est caractérisée par son type (numérique, alphabétique ou les deux mélangés) et sa longueur. La connaissance de sa longueur nous permettra d'allouer le nombre d espaces nécessaires après chaque libellé du masque. Le type de la zone nous conduira à faire un contrôle des données, caractère par caractère au fur et à mesure de leur introduction et à refuser les caractères non conformes (contrôle de numéricité). Nous serons donc amenés a créer deux tables supplémentaires : la table des longueurs (dénommee LZ par exemple) contenant la longueur de chaque zone, et celle des types indiquant le type de chacune (de nom TZ par exemple). Pour rappeler à l operateur quel type de zone il doit introduire, nous pourrons remplacer les espaces qui suivent les libellés par des « » (zones alphanumériques) et des «-» (zones numeriques)

Lors de la saisie des informations, l'operateur positionnera le curseur après chaque libellé et frappera la valeur des zones.

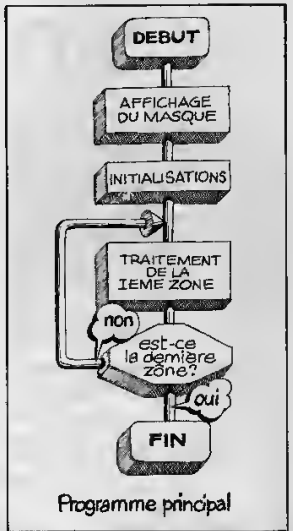
Il est pratique qu'apres l'introduction d'une zone (terminée par un retour chariot par exemple), le curseur saute automatiquement au début de la zone suivante. Il est également utile, au momant de la saisie, de pouvoir revenir sur des zones arrières où des erreurs de saisie auraient été commises. Cette fonction peut être réalisée par la pression de la touche *curseur gauche*.

Voici l'organigramme donnant les différents étages de notre sai-

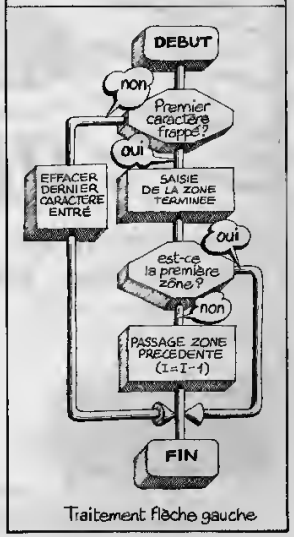


Principes de la saisie et tables utilisées

```
10 REM **** SAISIE D'ECRAN SUR TRS-80 ****
20 REM AUTEUR: J. BOISGONTIER
30 REM COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
40 POKE 16553,255        :REM *** SPECIAL TRS-80 ***
50 CLEAR 500
60 GOSUB 1000            :REM RESERVATIONS ET INITIALISATIONS
70 GOSUB 100             :REM *** S A I S I E ***
80 END
90 REM ----------------------------------------
100 CLS                  :REM *** SOUS PROGRAMME DE SAISIE ***
110 CLS                  :REM AFFICHAGE DU MASQUE
120 FOR I=1 TO NZ
130   XY=YY(I)*64+XX(I):REM CONVERSION X ET Y EN COORDONNEE @
140   PRINT@XY,NZ$(I)," :",
150   ON TZ(I) GOSUB 500,570 :REM CHOIX DU REMPLISSAGE DES ESPACES
160 NEXT I
170 I=1:CH$=""           :REM REMPLISSAGE DES ZONES
180 LG=LEN(NZ$(I))
190 XY=YY(I)*64+XX(I)+LG+2
200 SET((XX(I)+LG+1)*2,YY(I)*3) :REM POSITIONNEMENT DU CURSEUR
210 PRINT@XY,">",
220 X$=INKEY$: IF X$="" THEN 220
230 IF ASC(X$) = 13  THEN  IF CH$="" THEN 320 ELSE 300
240 IF ASC(X$) = 8 THEN
      IF CH$="" THEN IF I=1 THEN 220 ELSE RESET((XX(I)+LG+1)*2,YY(I)*3):
                     I=I-1:GOTO 180
                ELSE CH$=LEFT$(CH$,LEN(CH$)-1):PRINT X$,:GOTO 220
250 ON TZ(I) GOSUB 3000,3050   :REM VERIFICATION DU TYPE
260 IF Q=2 THEN 220            :REM ERREUR SI TYPE NON CONFORME
270 CH$=CH$+X$                 :REM CUMUL DANS ZONE
280 PRINT X$,                  :REM AFFICHAGE CARACTERE
290 IF LEN(CH$)<LZ(I) THEN 220 :REM ENCORE DES CARACTERES A LIRE
300 RE$(I)=CH$
310 CH$=""
320 IF I<NZ THEN RESET((XX(I)+LG+1)*2,YY(I)*3): I=I+1: GOTO 180
330 RETURN
500 REM ----------------------------------------
510 REM  INDICATEURS VISUELS DES TYPES DE ZONES ..
520 REM ALPHANUMERIQUES
530 FOR K=I TO LZ(I)
540   PRINT".",
550 NEXT K
560 RETURN
570 REM NUMERIQUES
580 FOR K=1 TO LZ(I)
590   PRINT"-",
600 NEXT K
610 RETURN
1000 REM ----------------------------------------
1010 REM ACQUISITION DU MASQUE
1020 RESTORE
1030 READ NZ
1040 DIM NZ$(NZ),XX(NZ),YY(NZ),LZ(NZ),TZ(NZ),RE(NZ)
1050 FOR I=1 TO NZ
1060   READ NZ$(I)        :REM LECTURE DES LIBELLES
1070 NEXT I
1080 FOR I=1 TO NZ
1090   READ XX(I)         :REM LECTURE DES ABSCISSES
1100 NEXT I
1110 FOR I=1 TO NZ
1120   READ YY(I)         :REM LECTURE DES ORDONNEES
1130 NEXT I
1140 FOR I=1 TO NZ
1150   READ LZ(I)         :REM LECTURE DES LONGUEURS
1160 NEXT I
1170 FOR I=1 TO NZ
1180   READ TZ(I)         :REM LECTURE DES TYPES
1190 NEXT I
1200 RETURN
3000 REM ----------------------------------------
3010 REM TESTS SUIVANT TYPES DES ZONES ...
3020 REM ALPHANUMERIQUES
3030 IF ASC(X$) < 32 THEN Q=2 ELSE Q=1
3040 RETURN
3050 REM NUMERIQUE
3060 IF ASC(X$) < 48 OR ASC(X$) > 57 THEN Q=2 ELSE Q=1
3070 RETURN
4995 REM ----------------------------------------
4996 REM DESCRIPTION DU MASQUE
4999 REM NOMBRE DE ZONES
5000 DATA 5
5005 REM LIBELLES
5010 DATA NOM,PRENOM,   RUE, VILLE, CPOST
5015 REM ABSCISSES
5020 DATA  10,    20,    10,    20,    10
5025 REM ORDONNEES
5030 DATA   5,     7,     9,    11,    13
5035 REM LONGUEURS
5040 DATA  30,    10,    20,    20,     5
5045 REM TYPES
5050 DATA   1,     1,     1,     1,     2
```



Programme principal



Traitement flèche gauche

Première parution : L'Ordi n° 14

sie. Le programme principal affiche tout d'abord le masque qui décrit l'enregistrement à saisir. Puis il réalise une boucle exécutée autant de fois qu'il y a de zones dans le fichier.

Pour chacune d'entre elles on effectue les opérations suivantes

— *positionnement du curseur après le libellé de cette zone*

— *chaque caractère est saisie au vol par balayage du clavier.*

Le retour au caractère précédent s'effectue par l'emploi du curseur gauche. Si on utilise ce curseur en début de zone, on revient à la zone précédente (sauf si on se trouvait positionné sur la première zone !) chaque caractère est soumis à un test qui dépend du type de la zone. S'il est erroné, il n'est pas pris en considération. Sinon il est affiché à l'écran et ajouté à une zone de cumul. Lorsqu'on a rentré assez de caractères pour remplir la zone, il y a saut automatique à la zone suivante. Le saut peut être également provoqué par l'introduction d'un retour chariot.

Une fois saisie, la valeur de chaque zone est stockée dans un tableau.

Les principes que nous venons d'énoncer servent à réaliser un programme de saisie. Nous l'avons écrit sur un TRS 80 et en donnons ci-dessous la liste intégrale.

Ce programme utilise certaines particularité du BASIC niveau 2. La saisie des caractères s'effectue au moyen de la fonction INKEY$. L'affichage en un point de l'écran est réalisé avec PRINTC. Cette dernière fonction n'utilisant qu'une coordonnée, il est nécessaire de convertir les valeurs (X, Y) en une seule valeur XY (ligne 190). La zone en cours de saisie est pointée par un caractère graphique affiché par l'instruction SET. Au passage à une autre zone, il est éteint par RESET.

Au début du programme, la description du masque est fournie par des données situées en DATA (lignes 5 000 à 5 050). Puis on exécute le sous-programme de saisie. A noter qu'il doit normalement être suivi d'un sous programme d'enregistrement sur mini-disquette ou sur cassette.

Le sous-programme affiche tout d'abord le masque (lignes 100 à 160). Puis les zones sont remplies avec les possibilités de correction d'erreurs et de contrôle de validité déjà évoquées (lignes 170 à 330). Chaque fois qu'une zone est remplie, elle est stockée dans un tableau (ligne 300).

Dans notre programme, les caractéristiques du masques sont stockées en DATA. Il serait intéressant de réaliser un programme pour créer ce masque. Par exemple on pourrait dessiner à l'écran l'image du masque telle qu'elle est affichée au début de notre programme ; puis, à partir de cette image, remplir les tableaux descriptifs du masque. En allant plus loin, un nom pourrait être affecté à chaque masque et sa description stockée sur mini-disquettes. Lors de l'utilisation d'un fichier il serait ainsi possible de charger en mémoire le masque qui lui est associé. (D'ailleurs plusieurs masques pourraient être associés à un même fichier).

La méthode de saisie que nous venons d'exposer devrait rendre de grands services pour les applications de gestion. Elle est facilement adaptable à d'autres ordinateurs individuels. Il suffit pour cela de modifier les parties du programme qui réalisent le calcul de la position des zones à l'écran et leur affichage suivant le mode d'adressage dont on dispose.

Jacques Boisgontier

Votre micro-ordinateur a besoin

CBM 64
VIC 20
ZX 81
SPECTRUM

DE NOUVEAUX LOGICIELS

Vous les trouvez chez

RUN

le spécialiste des logiciels de jeux... du sérieux... des extensions

Pour recevoir notre documentation, précisez votre matériel, découpez et renvoyez cette annonce à :

RUN INFORMATIQUE
62, rue Gérard, 75013 PARIS
Tél. : (1) 581.51.44

M ____
Adresse : ____
Matériel ____

12 83

pyc publicité

Référence 113 du service-lecteurs (page 172)

REIMS

SINCLAIR : ZX 81 - Spectrum
Extensions MEMOTECH 16-32-64 K
HAUTE RÉSOLUTION-PROGRAMMES, etc.

ORIC : 16 K - 48 K
COULEUR - SON - HAUTE RÉSOLUTION.

EPSON : HX 20 : 16 K - 32 K
ORDINATEUR PORTATIF AUTONOME

VICTOR, CASIO etc...

SYSTÈMES PROFESSIONNELS
COMPTABILITÉ - STOCK
PROGRAMMES SUR MESURE...

nous consulter !

HERCET MICRO-INFORMATIQUE
(26) 82.57.98 70, rue du Barbâtre-REIMS

Référence 112 du service-lecteurs (page 172)

faites varier la longueur de vos enregistrements

pour gagner de la place et du temps

L'enregistrement d'un fichier de données pose des problèmes liés aux temps d'accès et, notamment avec les cassettes, à la fiabilité de l'enregistrement. Ce dernier point nécessite souvent une modification du matériel. Le premier, par contre, peut être résolu par programme. En effet, la manière dont les informations sont lues ou écrites fait varier considérablement la rapidité d'exécution des opérations.

Le problème des fichiers, de leur création et de leur exploitation, est fondamental pour les utilisations professionnelles d'un ordinateur individuel. Le problème a été soulevé dans l'article « *Utilisations professionnelles : est-ce seulement un problème de matériel* » de Thérèse Rieul (*)

Nous envisageons d'expliquer ici le choix de l'enregistrement à longueurs variables, d'entrer dans le technique de base, de définir un sous-programme d'écriture et un sous-programme de lecture des enregistrements de ce type, et enfin d'appliquer ceci à la création de fichiers d'adresses permettant les transferts sur étiquettes auto-collantes (**)

Signalons que l'enregistrement ci-dessous est appelé *article*, il fait partie du *fichier*-adresses (composé de tous les articles). L'article adresse-complète est subdivisé en une *rubrique* NOM (la rubrique est un ensemble à signification précise, le nom d'une personne ici !), une rubrique prénom, adresse, numéro et localité.

La rubrique est subdivisée en *mots*, ensemble de *caractères* (lettres, chiffres ou signes) qui constituent la plus petite unité d'information exploitable par l'homme).

Par exemple : avenue des Tilleuls est la 3e rubrique de l'article X du fichier-adresses : cette rubrique est composée de 3 mots dont le premier (Avenue) est composé de 6 caractères alphabétiques.

Remarques : une information peut être représentée par un caractère alphabétique (AVENUE), numérique (887,75) ou alphanumérique (EDITEST 230589 F). (*)

Une chaîne de caractères étant constituée au maximum de 255 (**) caractères, il s'agit d'exploiter au maximum la longueur disponible. L'espace étant considéré comme un *caractère* blanc, il faut limiter les espacements entre données. C'est pourquoi il est préférable de représenter l'enregistrement (1) par l'enregistrement (2)

```
NOM.............PRENOM.........RUE............N°..LOCALITE.....
```

```
NOM*PRENOM*RUE*N°**LOCALITE
```

En effet, deux avantages peuvent être dégagés :

(1) Le deuxième enregistrement est plus court car le DESSIN DE ZONES doit réserver *plus* d'emplacements-caractères que la *longueur moyenne* des rubriques (nom, prénom, rue, n° et localité). S'il ne le fait pas, il y a des risques de tronquer un nom plus long ou une localité plus longue que la moyenne de données, etc., on va 2 fois plus vite que si l'on a 1/2 seconde de donnée, 1 seconde d'arrêt marche, 1/2 seconde de donnée, etc. Tout au moins bien sûr lorsque le temps de traitement de l'unité centrale est faible par rapport à la durée d'une lecture sur fichier magnétique — cassette ou disquette —, ce qui est le plus souvent le cas.

(2) Nous pouvons dire que la représentation (2) permet l'enregistrement de rubriques de longueurs quelconques (sans avoir à se préoccuper de la longueur du dessin de zone !). Il y a utilisation optimale de la mémoire de l'ordinateur et l'enregistrement sur un support magnétique ne demande qu'un minimum d'espaces arrêt-marche (on appelle *espace arrêt-marche* l'espace vide entre deux blocs successifs, en anglais *gap* = trou).

Lorsque les enregistrements sont courts et que la machine autorise des blocs d'une longueur supérieure (255 caractères), on a presque toujours intérêt à grouper plusieurs enregistrements par bloc, de manière à économiser de la place sur la bande magnétique, et à augmenter la vitesse d'exploitation.

En effet, un espace arrêt-marche dure un certain temps, fixe, par exemple 1 seconde. Si l'on a sur la bande 2 secondes de données, 1 seconde d'arrêt marche, 2 secondes

Une marque de rubrique est un caractère alphanumérique utilisé comme séparateur des rubriques d'un article. Nous ne considérerons que des articles composés d'un

W	U	E	S	T	E	N	S	E	R	CHS

Exemple : dessin de zone « NOM » avec 10 positions (la moyenne accordée), un nom de 13 lettres est évidemment tronqué

(*) L'Ordinateur Individuel *n° 8, p. 24-28.*
(**) voir page 95.

(*) *Ça, c'est le télex de l'OI.*
(**) Sur la plupart des BASIC des PSI habituels

même nombre de rubriques. Les mots d'une rubrique ne sont « séparés » que par la marque 'blanc' ou 'espace'.

```
110 DATA PAUL      EMILE    RUE DE LA GARE   123 76543 VILLE
120 DATA PIERRE    MICHEL   HAMEAU            9   98765 VILLAGE
```

marque de mot = espace

```
150 DATA PAU4L*EMILE*RUE DE LA GARE*123*76543 VILLE
160 DATA PIERRE*MICHEL*HAMEAU*9*98765 VILLAGE
```

marque de rubrique = astérisque

Comment le programme peut-il ordonner le traitement de ces chaînes de caractères ?

Une telle méthode de groupage des articles est pour l'homme difficile, tant à écrire qu'à déchiffrer. Il s'ensuit que les données de base sont représentées sur des grilles prévoyant l'information d'encombrement maximum. Ce n'est qu'à l'entrée des données dans l'ordinateur qu'un sous-programme spécial éliminera les zones non-employés : c'est le *programme de condensation* dont l'objet est d'augmenter la densité de l'information rangée en mémoire. Cf ci-dessous (*).

```
1 CLEAR 1000 : DEFINT A : DEFSTR L-S
2 GOSUB 1000
3 PRINT "FIN"
4 END
1000 REM *** SOUS PROGRAMME DE CONDENSATION ***
1010 REM *** COPYRIGHT WUESTENBERGHS DANIEL ***
1020 REM ***   ET L'ORDINATEUR INDIVIDUEL   ***
1030 INPUT "NOM :" ; NM
1040 INPUT "PRENOM :" ; PR
1050 INPUT "RUE, AVENUE, BOULEVARD,... :" ; R
1060 INPUT "NUMERO DE L'IMMEUBLE :" ; NO
1070 INPUT "LOCALITE :" ; LO
1080 S1=NM+"*"+PR+"*"+R+"*"+NO+"*"+LO+"**"
1090 A1=LEN(S1) : A2=LEN(S)
1100 IF A1+A2<255 THEN S=S+S1 ELSE PRINT#-1,S : S=S1
1110 INPUT "TAPEZ '@' LORSQUE VOUS AVEZ
TERMINE LA SAISIE DES ADRESSES" ; SO
1120 IF SO="@" THEN PRINT#-1,S : RETURN

1130 GOTO 1030
```

Le programme ci-dessus utilise la routine de condensation. L'instruction 1 080 compose l'article condensé, en 1090, la longueur de l'article est calculée ainsi que la longueur du bloc (A2). Si la cumul de ces deux longueurs dépasse 255 caractères, il n'y a pas addition des deux séquences mais enregistrement du bloc tel qu'il est (PRINT = # 1,S) et le nouveau bloc S est initialisé à la valeur du dernier article S1 (ELSE de l'instruction 1100). Les instructions 1110 à 1130 permettent de retourner au programme-maître si SØ = C, sans quoi l'écriture des articles continue (GOTO 1030).

Inversement, lorsqu'il faut imprimer le contenu de ce fichier-adresses, un *sous-programme de décondensation* rétablira la mise en page initiale, en analysant les différents blocs pour les décondenser en articles (les marques d'articles sont bien utiles à ce moment-là, et, reconnues par la machine, elles permettront la séparation des articles) et en rubriques (les marques de rubriques permettront la même procédure). Il suffira de cadrer le contenu des rubriques dans le dessin d'imprimé choisi.

Voici (p. 47) une routine de décondensation-lecture des blocs. Le programme est composé de 3 grandes parties (voir les REM).

(1) Lecture de la chaîne de caractères condensés ou bloc d'enregistrement sur cassette (ou en DATA, ou sur disquette, en changeant l'instruction 1211) (lignes 1000 à 1030).

(2) Lecture des articles de la chaîne (instruction 1040 à 1210). Nous utilisons la fonction MID$ (variable, p,n) qui donne les n caractères d'une chaîne dont le premier caractère est le (p) ième de la chaîne originelle.

(3) Lecture des rubriques de chaque article (il s'agit en fait d'un sous-programme du (2) ci-dessus — instructions 2000 à 2110).

Remarque. comme les blocs ne sont pas nécessairement de 255 caractères, la longueur du bloc doit être calculée par l'instruction LEN (bloc), la boucle ne lira donc que les (LEN (bloc) premiers caractères (instructions 1050 at 2020).

De l'instruction 1340 à l'instruction 1400, l'article appelé M1 est balayé en lecture afin de déceler la présence d'un astérisque. Supposons que le 8e caractère en soit un, nous aurons donc :

MID$ (M1, 8,1) = "*"
Y4 = 8-1 = 7
R = MID (M1, 1,7)

ce qui signifie que la variable R correspondra à la partie de M1 partant du 1er caractère jusqu'au 7e. Et Y3=7 + 2 = 9 car si la 1ère rubrique est terminée au 7e caractère, le 8e étant la marque, la lecture ne doit reprendre qu'au 9e !

Exemple d'exécution

Si la variable alphanumérique entrée (INPUT K7) est :

```
PAUL*EMILE*RUE DE LA GARE*123*45678 VILLE*PIERRE*MICHEL*HAMEAU*9*98765 VILLAGE*..MARC*ANTOINE*GRAND-RUE*8*76543 BOURGADE**
```

soit 5 adresses condensées.

La décondensation de ce bloc donnera, après décondensation, 5 articles de 5 rubriques :

```
PAUL*EMILE*RUE DE LA GARE*123*45678 VILLE*
PAUL
EMILE
RUE DE LA GARE
123
45678 VILLE
PIERRE*MICHEL*HAMEAU*9*98765 VILLAGE*
PIERRE
MICHEL
HAMEAU
9
98765 VILLAGE
LUC*ARMAND*RUE DU CENTRE*4*32109 BOURG*
LUC
ARMAND
RUE DU CENTRE
4
32109 BOURG
MARC*ANTOINE*GRAND-RUE*8*76543 BOURGADE*
MARC
ANTOINE
GRAND-RUE
8
76543 BOURGADE
```

Remarques

(1) Les marques utilisées (astérisque ici) ne doivent pas se rencontrer dans le texte (faute de quoi le texte sera considéré comme marque de mot et il y aurait confusion sur le nombre d'articles ou de rubriques).

(2) Les marques d'articles et de rubriques doivent toujours terminer l'article ou la rubrique puisqu'elles les délimitent (dans le bloc 2 astérisques après RIVIERE et dans une rubrique 1 astérisque après VILLE par exemple).

(3) La ligne 1310 est ici incorrecte (arrête l'exécution), mais facilite la manipulation du programme.

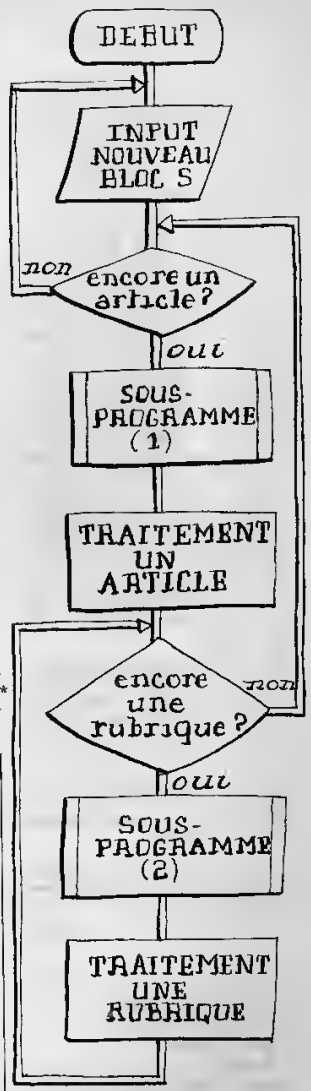
Le programme de synthèse de la page 48 reprend les sous-pro-

(*) *Tous les programmes sont donnés ici pour un TRS 80 niveau II utilisant une cassette*
- *CLEAR 1 000 réserve 1 000 caractères pour les chaînes*
- *DEFINT A tout nom de variable commençant par A désigne une variable de type entier*

DEFSTR L S tout nom de variable commençant par L M S désigne une variable de PRINT #1 écrire sur la cassette

Liste de la routine de décondensation - lecture des blocs

```
1 CLEAR 5000 : DEFINT A, X-Z : DEFSTR L-S
2 GOSUB 1200
3 PRINT "FIN"
4 END

1200 REM *** SOUS PROGRAMME DE DECONDENSATION ***
1210 REM *** AUTEUR :   DANIEL   WUESTENBERGHS ***
1220 REM *** COPYRIGHT:L'ORDINATEUR INDIVIDUEL ***
1230 REM ***        ET L ' AUTEUR              ***

1240 PRINT "CASSETTE EN POSITION DE LECTURE"
1250 INPUT#-1,S : PRINT S
1260 REM *** ROUTINE DE LECTURE DES ARTICLES ***
1270 Y1=1
1280 FOR X=1 TO LEN(S)
1290    IF MID$(S,X,2)="**" THEN 1300 ELSE 1340
1300    Y2=X
1310    M1=MID$(S,Y1,Y2-Y1+1)
1320    GOSUB 1360
1330    Y1=Y2+2
1340 NEXT X
1350 STOP : GOTO 1250
1360 REM *** ROUTINE DE LECTURE DES RUBRIQUES ***
1370 Y3=1
1380 FOR Z=1 TO LEN(M1)
1390    IF MID$(M1,Z,1)="*"  THEN 1400  ELSE 1440
1400    Y4=Z-1
1410    R=MID$(M1,Y3,Y4-Y3+1)
1420 PRINT R
1430 Y3=Y4+2
1440 NEXT Z
1450 RETURN
```



Organigramme normalisé du traitement d'un bloc

La liste du programme complet en BASIC

```
10 REM *** PROGRAMME COMPLET DE CONDENSATION ***
20 REM *** DECONDENSATION ET TRANSETIQUETAGE ***
30 REM *** AUTEUR :   DANIEL   WUESTENBERGHS ***
40 REM *** COPYRIGHT:L'ORDINATEUR INDIVIDUEL ***
50 REM ***        ET L ' AUTEUR              ***

60 CLEAR 5000 : DEFINT A-C, X-Z : DEFSTR L-S : DIM F(10,10)
70 FI=-1
80 LL="%                          %     %
90 S=""
100 PRINT "DESIREZ-VOUS ENREGISTREN UNE ADRESSE (1)"
110 PRINT "          OU IMPRIMER DES ETIQUETTES (2)"
120 PRINT "          OU ARRETER                 (3)"
130 INPUT "          VOTRE CHOIX " ; CH
140 ON CH GOSUB 1000 , 1200 , 160
150 GOTO 90
160 END

1000 REM *** SOUS PROGRAMME (1) DE CONDENSATION
1010 INPUT "NOM :" , NM
1020 INPUT "PRENOM :" , PR
1030 INPUT "RUE, AVENUE, BOULEVARD,... :" ; R
1040 INPUT "NUMERO DE L'IMMEUBLE :" ; NO
1050 INPUT "LOCALITE :" , LO
1060 S1=NM+"*"+PR+"*"+R+"*"+NO+"*"+LO+"**"
1070 A1=LEN(S1) : A2=LEN(S)
1080 IF A1+A2<255 THEN S=S+S1 ELSE PRINT#FI,S : S=S1
1090 PRINT "TAPEZ '@' LORSQUE VOUS AVEZ
TERMINE LA SAISIE DES ADRESSES"
1100 U$=INKEY$ : IF U$="" GOTO 1100
1110 IF U$="@" THEN 1120 ELSE 1010
1120 PRINT#FI,S : PRINT#FI,"FIN" : RETURN

1200 REM *** SOUS PROGRAMME (2) DE DECONDENSATION ***
1210 INPUT#FI,S
1215 IF S="FIN" THEN 1390
1220 PRINT "LECTURE DE L'ENREGISTREMENT"
1230 REM *** ROUTINE DE LECTURE DES ARTICLES ***
1240 Y1=1 : B=0
1250 FOR X=1 TO LEN(S)
1260    IF MID$(S,X,2)="**" THEN 1270 ELSE 1310
1270    Y2=X : B=B+1
1280    M1=MID$(S,Y1,Y2-Y1+1)
1290    GOSUB 1400
1300    Y1=Y2+2
1310 NEXT X
1320 FOR X1=1 TO B STEP 3
1330    FOR X2=1 TO C
1340       LPRINT USING LL, F(X1,X2), F(X1+1,X2), F(X1+2,X2)
1350    NEXT X2
1360    LPRINT "  " : LPRINT "  " : LPRINT "  "
1370 NEXT X1
1380 GOTO 1210
1390 RETURN

1400 REM *** ROUTINE DE LECTURE DES RUBRIQUES ***
1410 Y3=1 : C=0
1420 FOR Z=1 TO LEN(M1)
1430    IF MID$(M1,Z,1)="*"  THEN 1440  ELSE 1480
1440    Y4=Z-1 : C=C+1
1450    R=MID$(M1,Y3,Y4-Y3+1)
1460 F(B,C)=R
1470 Y3=Y4+2
1480 NEXT Z
1490 RETURN
```

grammes de condensation et de décondensation. Le choix est opéré en début de programme (instruction 90) et l'instruction 100 envoie au sous-programme choisi (écrire ou condenser, lire ou décondenser). L'instruction 61 définit le *format d'impression* qui sera utilisé par l'instruction 1303. L'instruction 1 305 imprime trois lignes blanches (ce sont les interlignes).

On pourrait ajouter une instruction 1212 IF S = " FIN " THEN RETURN qui renverrait à l'instruction 100 comme cela est fait à 1150.

L'organigramme reprend le traitement de décondensation d'un bloc d'anregistrement (instructions 1200 à 1410).

Certaines améliorations peuvent être apportées à la présentation du programme (PRINT "libellés d'instructions pour l'utilisateur"), ces instructions ont été supprimées pour ne pas alourdir le listing.

Le lecteur modifiera facilement le programme afin d'arriver à une utilisation spécifique de ce qu'il a appris pour les adresses.

Daniel Wuestenberghs

PAULEMILE*RUE DE LA GARE*123%45678 VILLE**PIERRE*MICHEL*HAMEAU*234*98765 VILLAGE**LUC*ARMAND*_____ON BOURG**MARC*ANTOINE*GRAND-RUE*8*76543 BOURGADE**LOUIS*LEOPOLD*PETITE RUE*21*24680 R

Voici une illustration du principe du programme. Ci-dessus : les enregistrements sous forme condensée. A droite : les mêmes sous forme décondensée.

PAUL	PIERRE	LUC
EMILE	MICHEL	ARMAND
RUE DE LA GARE	HAMEAU	RUE DU CENTRE
123	234	4
45678 VILLE	98765 VILLAGE	32109 BOURG
MARC	LOUIS	ALPHONSE
ANTOINE	LEOPOLD	ROBERT
GRAND-RUE	PETITE RUE	RUE DROITE
8	21	10
76543 BOURGADE	24680 RIVIERE	12345 MONT

organisez vos données en listes linéaires doubles

Le programme en Basic que nous publions illustre l'organisation de données appelée *liste linéaire double*. Cette structure de liste linéaire double donne la possibilité, par exemple, de créer des index ou d'organiser de façon performante un fichier « commandes ». Elle permet aussi d'assurer la gestion d'un stock de produits finis. Découvrons donc cet outil logique puissant.

Intéressons-nous d'abord aux structures classiques d'organisation. Il faut préciser que si, par commodité, les arguments sont, dans le programme et le corps de l'article, des nombres entiers, dans la réalité les « valeurs » comparées sont des arguments numériques ou alphanumériques qui peuvent être eux-mêmes constitués d'éléments ordonnés de « majeur » à « mineur ».

Structures comparées

1 - Séquentielle Triée -

	(1)	(2)	(3)	(4)	(5)	
Indices						
VALEURS	125	342	343	768	802	

2 - Adressée ou "Pointée"

	(1)	(2)	(3)	(4)	(5)	
Indices						
VALEURS	768	342	802	125	342	
Pointeur	[4]	[5]	[2]	[1]	[3]	

3 - Liste Simple (chaînage vers le Successeur)

	(1)	(2)	(3)	(4)	(5)	
Indices						
VALEURS	768	343	802	125	342	
Chaînons	3	1	0	5	2	

Début en -> (4)

On peut comparer rapidement ces trois structures.

L'organisation séquentielle est peu gourmande en place ; elle est rapide à lire et elle permet l'emploi de la recherche dichotomique. En revanche, elle est difficile et coûteuse à réorganiser (recopies ou décalages en cascade lors des additions et suppressions).

L'organisation adressée a le grand avantage d'être beaucoup plus performante que la précédente lors des réorganisations, celles-ci ne portant que sur la table – ou le fichier – des « pointeurs ». Il va de soi que le fichier des pointeurs est physiquement séparé du fichier principal. La place occupée est un peu plus grande. La recherche dichotomique est possible.

La liste simple présente un inconvénient : la recherche dichotomique n'y est pas possible. On lit le fichier ainsi :

a) consultation de la racine de la liste ; on a ainsi le rang physique (indice) de l'élément le plus faible (ou considéré comme étant le premier de la liste) ;

b) on progresse ensuite de proche en proche (et le plus souvent en zig-zag), éventuellement jusqu'en butée, c'est-à-dire jusqu'à l'élément qui n'a pas de successeur (valeur du chaînon = 0)

Notons que, contrairement à

Première parution L OI n° 46

l'organisation adressée, las chaînons peuvent, ici, faire partie de l'enregistramant principal. En effet, une mise à jour (addition d'un élément par axample) ne donne lieu qu'à un minimum « d'écriture » :

inscription dans la rubrique « successeur » du nouvel enregistrement, de la valeur du rang (ou indice) de l'élément suivant

modification de la valeur du chaînon « successeur » de l'élément précédent.

Malheureusement, la liste simple présanta d'autras inconvénients : on ne peut la lire que dans un saul sens at, dans le cas d'una racherche avant la mise à jour, il faut, da façon continue, mémoriser la dernier « précédant » (c'est-à-dire son indice), ce qui pénalisa la vitessa de recherche

Voyons maintanant la liste double (encadré ci-après)

Lista double

(1)	(2)	(3)	(4)	(5)	← Indices
768	343	802	125	342	VALEURS
3	1	0	5	2	Success.
2	5	1	0	4	Préced.

Début en (4) (Racine des successeurs)
Fin en (3) (Racine des Précédents)

Pour lira an ordre « croissant », on utilise les « successaurs », et si le nom de l'élément ast V(i), on aura successivemant V(4), V(5), V(2), V(1) et V(3) L'utilisation des « précédents » est évidemmant analogue.

Cela nous amène à la raprésentation « logique » d'une liste double (encadré ci-dessous) :

Dans notre programma, las variables dans lesquellas sont entretanuas les valaurs des racines commencant par « D » pour les précédents, et par « F » pour les successeurs. Nous aurions par exemple : D1=4, et F1=3.

Occupons-nous maintenant du programme. Les spécifications qu'il raspecte sont décrites ci-après.

1) Gérer en temps ráel (lire, ajouter, supprimer) par liste double un ensemble de nombres entiars compris entre 1 et 999
2) Ces nombres seront en ordre croissant sur les successeurs (et donc en ordre décroissant sur les précédents)
3) Les nombres seront entrés au claviar, précédés de la commande « T » (Trouver), « A » (Addition), « S » (Suppression)
4) Les homonymes saront autorisés, mais pourront âtre interdits par une intervention simple du programmeur
5) Au départ la table sera vide, les valeurs principales étant à zéro.
6) Les règlas da mise à jour (en temps réel) sont les suivantes :
. un ajout doit se traduire par l'utilisation (at donc la suppression) du « trou » d'indice le plus bas ;
. si l'ajout a déjà des homonymes, il devra être chaîné avac ceux-ci de façon à respecter l'ordre des indices croissants. ;
. una suppression ne pourra évidamment avoir lieu que si un élément, au moins, est égal au nombre indiqué ; s'il y en a plusieurs, l'élément à supprimer sera celui de rang le plus élevé.
7) Las commandes supplémentaires suivantes seront programmées :
« V » (comme Voir) entraînara l'affichage da la tabla principale et das tablas de chaînage , si la dimansion de la table est supérieure à quinze, seuls les quinze premiers éléments seront affichés ; la valeur des racines sera également affichée
« L » (comme Lister) doit permettre de lister les valeurs principales dans leur ordre croissant ;
. « I » (comma Inverse) fera la même chosa en ordra décroissant.
. « F » (comme Fin) mettra fin au programma.

Représentation logique d'une liste double

(4)	(5)	(2)	(1)	(3)	Ind.(K)
125	342	340	768	802	VALEURS
5	2	1	3	0	Succ.
0	4	5	2	1	Préced.

Passons maintenant aux description et réalisation du programme

On voit que la routina d'initialisation « INIT » est placée en fin de programme. Cela permet de placer tout au début du programme les lignes de code qui donnent la structura générale et rejettant, à

Programme de listes linéaires doubles

```
130 'LISTES DOUBLES - Auteur : Bernard SESSE - "MAULIST/B" - 01/11/11
135 ESSE
140 GOSUB 135 -- INIT
145 GOSUB 410'-- SAISIE
120 : GOSUB 410'-- SAISIE
125 ON C GOSUB 135,265,315,440,545,580,615: GOTO 120
130 'TROUV, ADD, SUPP, VOIR, LIST, INV, FIN
135 : '--- TROUV ---
140 RX=R: DX=D1: FX=F1: AX=0: GOSUB 160: K1=KX: T1=TX: G1=GX'-- RECH
150 IF T1 THEN PRINT"Trouve Place =";K1 ELSE PRINT"Pas Trouve"
155 RETURN'-- RECH ---
160 IF DX=0 THEN KX=0: TX=0: GX=-1: RETURN'--> (vide)
165 IF DX=0 THEN KX=0: TX=0: GX=-1: RETURN'--> (vide)
170 '------ Chaînage sens rech.
175 AV=KX-V(DX): AF=V(FX)-KX
180 IF AV   AF THEN 230'-- recherche inverse
185 IF AV=AF AND AV 0 THEN IF AX=DX   FX AX THEN 230'-- suite homonymes
190 '------ Recherche (S) (directe)
195 AX=DX
200 IF AX   FX > V(KX) THEN KX=S(KX): GOTO 200
205 IF AX   V(KX) THEN TX=0: GX=-1: RETURN
210 : IF AX   KX THEN AX=S(FX): IF V(X)=RX THEN 210
215 TX=1: IF V(FX)=RX THEN GX=-1: RETURN
220 FX=(KX): GX=0: RETURN
225 '------ Recherche (P) (inverse)
230 :   KX=FX
235 :   IF KX   V(FX) THEN KX=F(FX): GOTO 235
240 IF KX   V(FX) THEN TX=1: GX=0: RETURN
245 TX=-1: IF V(FX)=RX THEN GX=0: RETURN
250 FX=S(FX): GX=-1: RETURN
260 RETURN
265 :
270 RX=0: DX=D0' FX=F0: AX=0: GOSUB 160: F0=KX: T0=TX: G0=GX'   RECH
275 IF NOT T0 THEN PRINT"PLAS DE PLACE": RETURN
280 RX=K: DX=D1: FX=F1: AX=0: GOSUB 160: R1=KX: T1=TX: G1=GX -- RECH
285 IF NOT T1 OR R1=T0 THEN 275
290 PRINT"Add Place =";R1;" Add Impos.";: RETURN
295 FX=R1: RX=0: DX=D0: FX=F0: GOSUB 385: D0 DX: F0=FX -- FTRSUFP
300 V(R)=S
305 FX=K1: DX=D1: FX=F1: GX=G1: RX=R0: GOSUB 355: D1=DX: F1=FX' -- FTRADD
310 RETURN
315 :   --- SUPP ---
320 RX=K:DX=D1:FX=F1:AX=32767: GOSUB 160: K1=KX: T1=TX: G1=GX -- RECH
325 IF NOT T1 THEN PRINT"PRINT"NON TROUVE SUPP.IMPOSS." RETURN
330 V(K)=0
335 KX=K1: DX=D1: FX=F1: GOSUB 385: D1=DX: F1=FX'-- FTRSUFP
340 RX=K: DX=D0: FX=F0: AX=0: GOSUB 160: K0=KX: T0=TX: G0=GX -- RECH
345 KX=K0: DX=D0: FX=F0: GX=G0: NX=K1: GOSUB 325: D0=DX: F0=FX -- FTRADD
350 RETURN
355 :   --- FTRADD ---
360 IF GX THEN H=FX: G=F(FX) ELSE H=KX: H=S(KX)
365 S(NX)=H: F(NX)=G: S(G)=NX: F(H)=NX
370 IF G=0 THEN DX=S(H): S(H)=0
375 IF H=0 THEN FX=F(H): F(H)=0
380 RETURN
385 :   --- FTRSUFP ---
390 H=S(KX): P=F(KX): S(P)=H: P(H)=P
395 IF P=0 THEN DX=S(H): S(H)=0
400 IF H=0 THEN FX=F(H): F(H)=0
405 RETURN
410 :   --- SAISIE ---
415 PRINT
420 : PRINT"T(rouver, A(jout, S(upp., V(oir, L(ister, I(nverse, F(in ?";
425 C$=INPUT$(1):PRINT C$
430 IF C$ = "T" THEN C=1: GOTO 470
435 IF C$ = "A" THEN C=2: GOTO 470
440 IF C$ = "S" THEN C=3: GOTO 470
445 IF C$ = "V" THEN C=4: GOTO 470
450 IF C$ = "L" THEN C=5: GOTO 470
455 IF C$ = "I" THEN C=6: GOTO 470
460 IF C$ = "F" THEN C=7: GOTO 470
465 GOTO 420
470 : R=0: INPUT"NUMERO R(echerche) ";R: IF R   1 OR R >999 THEN 470
475 : RETURN
480 : '--- VOIR ---
485 CLS:PRINT:PRINT
490 PRINTUS;"I  ";UN$;"FDR I=1 TO MIFPRINTUSING"######";V(I);:NEXT I:PRINT
495 PRINTUS;"V(I)";UN$;"FDR I=1 TO MIFPRINTUSING"######";V(I);:NEXT I:PRINT
500 PRINTUS;"S(I)";UN$;"FDR I=1 TO MIFPRINTUSING"######";S(I);:NEXT I:PRINT
505 PRINT:PRINTSTRING$(79,"-")
510 PRINTUS;"F(I)";UN$;"FDR I=1 TO MIFPRINTUSING"######";F(I);:NEXT I:PRINT
515 PRINTSTRING$(79,"-")
520 PRINTUS;"P(I)";UN$;"FDR I=1 TO MIFPRINTUSING"######";P(I);:NEXT I:PRINT
525 PRINTSTRING$(79,"-")
530 PRINTUS;"L(I)";UN$;"  ( D0=";D0;" F0=";F0;")   ( D1=";D1;" F1=";F1;")"
535 PRINTSTRING$(79,"-")
540 RETURN
545 : '--- LIST ---
550 PRINT
555 IF D1=0 THEN PRINT "Vide, desserrement vide";: GOTO 570
560 I=D1:PRINTSPC(4);
565 : PRINTUSING"######";V(I);:IF S(I)>0 THEN I=S(I):GOTO 565
570 : RINT
575 : RETURN
580 : '--- INV ---
585 IF D1=0 THEN PRINT "Vide, tres vide";:GOTO 605
590 I=F1:PRINTSPC(4)
595 : PRINTUSING"######";V(I);:IF F(I)>0 THEN I=F(I):GOTO 600
600 : PRINT
605 : '--- FIN ---
610 RETURN
615 : '--- FIN ---
620 GOSUB 480: GOSUB 545: GOSUB 580'-- VOIR, LIST, INV
625 PRINT:PRINT"AU REVOIR":END
630 RETURN
635 : '--- TRAIT ---
640 DL$: DEFINT A-Z
645 HY=1 '--- ( 0, homonymes non-autorises ), ( -1, autorises )
650 INPUT"LT";LT
655 IF LT>15 THEN MI=LT ELSE MI=15
660 DIM V(LT),S(LT),P(LT)
665 FOR I=1 TO LT: S(I)=I+1: P(I)=I-1: NEXT I: S(LT)=0
670 D0 :F1=LT: D1=0'--- (Racines Trous et Valeurs)
675 VT$=CHR$(26):UN$=CHR$(25) '--- ( Video Inverse & Video Normale )
680 RETURN
685 '******
```

Copyright L'Ordinateur Individuel et l'auteur

la fin, des instructions parfois nombreusas qui ne sont exécutées qu'una seula fois Cetta disposition augmente las performances de certains Basic.

Le programme se termine lors de l'exécution de la routine « FIN » au sein da laquelle se trouve une instruction « END ».

Regardons rapidament la routine « SAISIE » : (ici lignes 410 à 475). Rien de particulier, si ce n'est l'instruction INPUT$ (n) permettant, an Basic TRS-80 Modèla 2, de saisir « n » caractères (ici un), sans appuyer sur ENTER (RETURN).

Avant de voir la routine INIT, reconnaissons qu'un problème se pose : faut-il chainer les trous entre eux ? Puisqu'on peut les identifier à laur valeur nulle (zéro numérique), on pourrait les trouver par une banale recherche séquentiella. Remarquons simplement que, suivant l'emplacament et le nombra de cas trous, la durée de recharche peut varier fortament.

Nous avons choisi la solution du chainage. Il fallait donc, à l'initialisation, chainer (par les successeurs et las précédents) une table de valeurs nullas de longueur LT.

La routine INIT se trouve aux lignes 635-680

LT (Longuaur Tablas) permat de dimensionner la table V() et les tables da chainage S() et P(). Les lignas suivantes initialisent la table vide ainsi que la valeur des racines

Rappalons que les valeurs des racines représentent das indices Ainsi D0 (racina Début das trous) = 1 veut dire que le premier des trous est à la place 1.

D1 at F1 sont les racines de la liste des nombres ; leur valeur 0 indiqua qua cette liste ne commenca nulle part et qu'ella est, pour l'instant, vide.

Et si nous nous lancions dans la programmation ?

1) Commandas jugéas simplas (opinion toute personnalle, bian sûr !) : ce sont les commandas V(oir), L(ister), I(nvarse) et F(in).

La commande V(oir) pourrait cartainement âtre programmáa de façon plus fina.

Les commandes L(ister) et I(nverse) sont claires. C'ast la valeur 0 (zéro numérique) du successeur (ou du précédant) qui arrête le processus.

2) Les commandas T, A et S.
T(rouver) impliqua :
Chercher et randra compte.
A(ddition) implique :
Chercher un trou.
S'il n'y a aucun trou déclarer addition impossibla
Si on trouve au moins un trou *chercher* la position logique du nouveau nombre (si on trouve un nombre idantique, ne poursuivra que si l'homonymia est autorisée).
Supprimer logiquement le trou (modification des chainons suite à la suppression)
Installer physiquemant le nouveau nombra.
Ajoutar logiquement ce nombre (modification des chainons à la suite d'une addition).
S(upprimer) implique :
Chercher la nombra (s'il n'existe pas, annuler).
Suppression physique.
Suppression logique (chainons).
Chercher la place logique du nouveau trou.
Ajoutar logiquement le nouveau trou

Il ressort de catte analyse qua trois fonctions ont un rôle prépondérant :
Chercher, fonction assurée par « RECH » ;
Addition (ou AJOUT) logique assurée par « PTRADD » ;
Suppression logique traitéa par « PTRSUPP ».

Ces fonctions doivent être traitées par des sous-programmes, car on y fait appel en plusieurs endroits. Il apparaît, en outra, que ces sous-programmes auront à traiter, suivant le cas, des donnéas et donc des variables différentes. En effet, recherche, additions et suppressions se font soit dans la liste des valaurs, soit dans celle des « trous ».

On est donc amené à « passer des paramêtres » aux sous-programmes Le procédá est ici le suivant : les arguments, fournis au sous-programme et rastitués à l'exténeur, sont traités sous des noms de variablas propres à celui-ci : ici les variables « internes » se terminent par la lettra « X ».

La transmission des arguments se fait sur la même ligne que celle de l'instruction GOSUB Par exemple le sous-programme imaginaire

```
100 : === PROCESS ===
110 IF UX > BX THEN WX=UX · RETURN
120 IF UX = BX THEN WX=0 : BX=BX/2 · RETURN
130 WX=BX · RETURN
140 RETURN
```

pourra être appelé de la façon suivante :

```
640 UX=UA : BX=BA : GOSUB 100 : WA=WX : BA=BX' — — PROCESS
```

La programmation de RECH, PTRADD et PTRSUPP

La syntaxe d'appel de RECH (Recherche) est la suivante :
nnn RX= ? : OX= ? : FX= ? AX= ? : GOSUB yyy : ?=KX: ?=TX: ?=GX' — — RECH
où :
RX = valeur du nombra,
DX = racine successaurs (D comma Début),
FX = racine précédents (F comme Fin),
AX = adressa (indice à prendre en compta pour l'insertion logique),
KX = indice, soit du nombre trouvé, soit du nombre immédiatement supérieur ou inférieur
TX = 0 si non trouvé, et — 1 si trouvé
GX = 0 si le nombre charché est *logiquamant* à droite du nombre d'indica KX, et = — 1 si à gauche

Par exemple, on est an train d'exécuter une commande « A ». Une première recherche a indiqué qu'un emplacement disponible (trou) existait en 143. Nous avons à insérer le nombre 614. Les homonymes sont autorisés, donc possibles. Supposons que la situation avant insertion soit celle représentée ci-dessous.

Situation avant insertion

(27)	(92)	(19)	Indice
614	614	856	VALEUR
92	19	0	Succ.
?	27	92	Précéd.

Exemples d'exécution de listes linéaires

```
T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? U
I — 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
V(I) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
==========================================================
S(I) 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0
F(I) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LT 15 ( D0= 1 F0= 15 ) ( D1= 0 F1= 0 )
.....
T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? T
NERE R(echerche) ? 9

Pas Trouve

T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? S
NERE R(echerche) ? 1

NON TROUVE Sous-Impass

T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? L

Vide: desespérement vide

T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? I

Vide. tres vide
```

```
T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? S
NERE R(echerche) ? 224

T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? S
NERE R(echerche) ? 876

T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? A
NERE R(echerche) ? 17

T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? U
I — 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
V(I) 345 234 209 17 27 234 97 678 0 101 48 842 76 23 45
==========================================================
S(I) 3 4 2 14 15 1 13 12 0 3 7 0 10 5 11
F(I) 4 3 10 0 14 2 11 1 0 13 15 0 7 8 5
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LT 15 ( D0= 7 F0= 9 ) ( D1= 4 F1= 12 )
.....
T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? I

17 23 27 45 68 87 98 101 209 234 345 678 842

T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? I

842 678 345 234 209 101 98 87 48 45 27 23 17
```

```
T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? S
NERE R(echerche) ? 27

T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? A
NERE R(echerche) ? 63

T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? B
NERE R(echerche) ? 68

T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? S
NERE R(echerche) ? 68

T(rouver, A(jout, S(upr., V(oir, L(ister, I(nverse, F(in ? F
I — 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
V(I) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
==========================================================
S(I) ? 3 4 5 6 7 8 9 10 11 12 13 14 15 0
F(I) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LT 15 ( D0= 1 F0= 15 ) ( D1= 0 F1= 0 )
.....
Vide. tres vide
OU REVOIR
Ready
>SYSTEM"DUAL OFF
Dim1 Routine now OFF
```

L'indice (143) du trou à utiliser est dans la variable K0. La valeur (614) dans R. On écrit :

```
nnn RX=R :DX=01 :FX=F1
:AX=K0 :GOSUB « RECH »
:K1=KX :T1=TX :G1=GX.
```

Les valeurs produites seront : K1=92, T1=-1 (trouve), GX=0 (ce qui indique une insertion à faire entre « 92 » et son successeur).

Nous laisserons le soin au lecteur de décortiquer le sous-programme RECH. Notons cependant que le choix du « meilleur » sans de recherche (par les successeurs ou les précédents) permet non seulement de minimiser le nombre des itérations, mais aussi de supprimer le test de non-nullité du successeur ou du précédent.

Il y a lieu aussi de noter le rôle des adresses particulières, AX=0 et AX=32767.

Le module PTRADD est appelé pour modifier les chaînons en cas d'addition. Supposons que nous venions de sortir de RECH avec les nouvelles valeurs de K1 et G1 (Peu importe la valeur de T1, car nous supposerons que l'addition est autorisée.)

Procédons d'abord à la mise en place physique :

nnn V(K0)=R

Appelons H (comme Haut), l'indice du nombre successeur du nombre R à insérer et B (Bas) celui de son précédent.

Si G1=— 1 on se situe à gauche et donc H=K1 et B=P (K1).

Sinon on a (nous laissons au lecteur le soin de faire le dessin correspondant) : B=K1 et H=S (K1).

Cette question des variables intermédiaires H et B réglée, on peut écrire :
S (K0)=H
P (K0)=B
ce qui met à jour les pointeurs du nouveau nombre, et :
S(B)=K0
P(H)=K0
qui raccorde les voisins au nouveau venu.

La valeur K1 est connue dans le module sous l'appellation KX, celle de K0 sous celle de NX. On retrouve la programmation effectivement présentée.

Passons maintenant à la modification des valeurs des racines

Lorsqu'il y a insertion ou suppression d'une valeur « racine » celles-ci (ou l'une de celles-ci) sont à modifier. Le lecteur intéressé notera l'utilisation qui est faite des indices 0 des tables de chaînage S() et P().

Il faut remarquer que les valeurs de S(0) et/ou P(0) sont dans ces parties transférées dans DX (et donc dans D0, D1, D...) et/ou FX (et donc dans F0, F1, F. .) et que S(0) et/ou P(0) sont immédiatement remis à zéro.

Nous ne dirons rien de PTRSUPP, plus simple que PTRADD.

En fait nous ne parlerons plus de la programmation, si ce n'est pour dire un mot de l'utilisation des variables entières comme variables logiques. Il s'agit d'expressions comme : IF HY THEN..

La valeur qui correspond à FAUX est 0 (16 bits éteints). Si HY n'est pas nul, l'expression ci-dessus se comportera comme si HY était VRAIE.

Donnons à HY la valeur 1 et à JZ la valeur 2. Faisons IF HY AND JZ THEN ..

L'expression HY AND JZ est fausse (nulle). Nous pensons donc qu'il vaut mieux s'en tenir en conséquence à — 1 comme constante VRAI. Dans le cas du présent programme il aurait probablement mieux valu initialiser deux variables FA=0, VR=—1 dans INIT.

Divers développements sont possibles.

1) Améliorations.
Affichage.
Routine RECH ; l'améliorer d'une façon générale ; choisir le sens de lecture en fonction non pas de la moyenne mais de la médiane.

Mettre en place des compteurs donnant le nombre de places libres, le nombre de nombres (somme des 2 = LT), etc.

2) Utilisation des listes doubles.
On peut les employer pour tenir des index correspondant à tel ou tel critère de tri. Mais la recherche est toujours relativement longue, car elle est « séquentielle » (elle porte en moyenne sur le quart de la liste, lorsque le choix du sens de recherche a pour base une comparaison avec l'élément « médian »).

On peut l'utiliser aussi comme « magasin automatique » de données : supposons que nos nombres soient maintenant des numéros d'articles et que nous ayons environ deux cents clients pouvant avoir en moyenne vingt articles en commande. Cela fait donc environ quatre mille enregistrements.

Si nos clients sont numérotés 1, 2, 3 ... 200, et si nos « racines » sont des variables indicées, nous avons « immédiatement » accès à la sous-liste du client 45, car ses racines sont D (45) et P (45). La recherche séquentielle dans la sous-liste elle-même est alors de très courte durée.

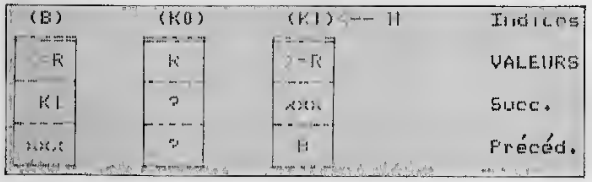
On peut aussi l'employer comme méthode générale d'accès : nous pouvons traiter des indicatifs par HASH-CODE pour obtenir un nombre qui soit l'indice des racines D() et F() à utiliser. Tous les nombres ayant les mêmes racines seront dans une même sous-liste et une lecture séquentielle (chaînée) permettra d'accéder à l'indicatif souhaité.

Nous ne nous étendrons pas davantage sur cette dernière possibilité. Nous la citons parce que c'est un cas de gestion de fichier sans réorganisation périodique nécessitée par les mises à jour.

Soulignons l'aptitude des listes doubles à la gestion d'emplacements vides mis en commun.

Cette aptitude peut être utilisée aussi bien pour la gestion d'emplacements physiques (par exemple dans les magasins automatiques de produits finis) que pour celle des fichiers informatiques.

Bernard Besse



une table d'index pour un accès plus direct

La plupart des petits systèmes individuels actuels disposent de fichiers permettant un accès aux données à partir d'un numéro d'enregistrement physique. Or, en particulier dans les applications de gestion, l'utilisateur désire accéder aux informations à partir d'une clé alpha-numérique (un nom ou un code par exemple), et de son numéro d'enregistrement. On peut, bien sûr, rechercher une information dans un fichier à partir d'un nom ou d'un code, en lisant séquentiellement tous les enregistrements. Mais compte tenu du temps d'accès à chaque enregistrement (de l'ordre de 100 ms), l'information cherchée n'est retrouvée qu'après plusieurs secondes pour un fichier de taille moyenne.

L'utilisateur se trouve donc confronté au problème suivant :

Comment établir la correspondance entre les clés d'accès utilisateur et le numéro d'enregistrement où sont rangées les informations, ceci en limitant au maximum le nombre d'accès au disque et en réduisant ainsi les temps d'accès ?

Deux principes sont généralement utilisés :
Soit le *Hash-Code* qui établit la correspondance par un *calcul d'adresse* de rangement effectué sur la *clé*.
Soit l'*accès indexé* qui établit la correspondance à l'aide d'une *table*.

Si la première méthode est plus simple dans son principe, elle est aussi plus coûteuse en *place disque*, du moins avec les fichiers classiques. C'est donc l'*accès indexé* que nous étudierons dans cet article.

De nombreuses méthodes existent, plus ou moins simples à programmer. Leur choix se fait en fonction des exigences de l'application et des particularités des fichiers utilisés.

Voici les principales, de la plus simple à la plus complexe :

L'accès indexé simple : on utilise une table en mémoire centrale, constituée en début de travail à partir du fichier.

L'accès indexé avec sauvegarde sur disque de la table d'index : la table des index, chargée en début de programme à partir du disque, est stockée sur disque en fin de travail de façon à permettre une réutilisation rapide lors d'un travail ultérieur.

L'accès indexé avec index continu en ordre croissant sur disque.

L'accès indexé avec index à 2 niveaux.

Examinons-les successivement.

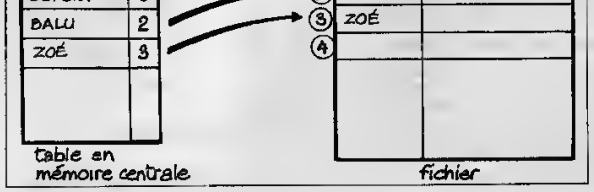
Accès indexé avec table de correspondance en mémoire centrale

Dans la version la plus simple de l'accès indexé, la table de correspondance est résidente en mémoire centrale et se constitue en début de travail par la lecture séquentielle de tout le fichier (schéma ci-dessous).

Lorsque cette table n'est pas triée, sa consultation est nécessairement séquentielle, mais relativement rapide si les clés ne sont pas trop nombreuses. Il suffit ensuite d'un accès disque pour retrouver l'enregistrement associé à la clé.

Mais plusieurs dizaines de secondes sont nécessaires à la constitution de la table si le fichier a une taille importante.

Aussi cette méthode n'est-elle guère exploitable dès que le nombre d'enregistrements physiques du fichier devient important, au-delà d'une centaine. Cependant, elle a l'avantage d'être simple à programmer, et aussi celui de s'adapter



Première parution : L'OI n° 16

aisément à une application existante par l'appel de 2 sous-programmes : un sous-programme de création de la table et un sous-programme de recherche dans le table

Si la table d'index est en ordre croissant, la recherche s'y fait par dichotomie et devient ainsi plus rapide. Voir le programme de recherche dichotomique page 115.

Il est possible de ne stocker dans la table d'index que les seules premières lettres des clés. Ceci contribuera à réduire la place mémoire occupée, et à diminuer les temps de comparaison de la clé cherchée avec les clés de la table. Mais en cas d'identité entre les premières lettres de différentes clés, plusieurs accès disques seront éventuellement nécesseires.

Lorsque les enregistrements physiques (repérés par un numéro) comportent plusieurs enregistrements logiques eyant chacun leur clé, on peut coder dans la table d'index l'ensemble du numéro physique et de la position dans l'enregistrement physique de la façon suivante :

X = (No physique*NLOG) + POSITION

où NLOG représente le nombre d'enregistrements logiques par enregistrement physique. Le décodage se fera par
No physique = INT (X/NLOG),
POSITION = X-INT (X/NLOG)

Voici un programme d'accès indexé avec table en mémoire centrale (ci-contre, en haut). La mise à jour de la table en temps réel, en cas d'ajout de clé, n'a pes été programmée sur cet exemple. C'est donc seulement à la prochaine constitution de la table, c'est-à-dire lors du prochain travail, que les clés ejoutées y seront présentes.

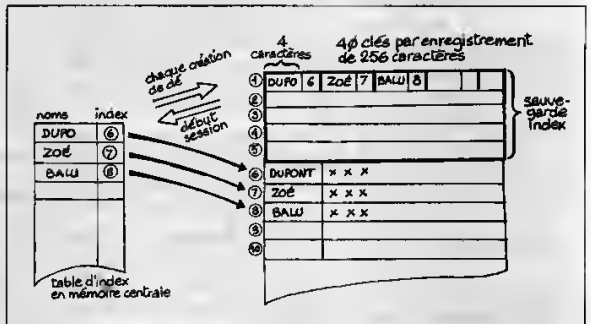
Accès indexé avec sauvegerde sur disque de le table d'index

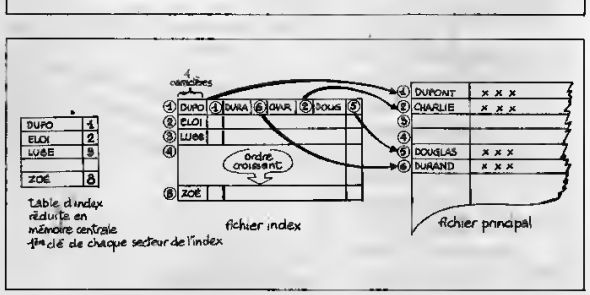
Une sauvegarde de la table d'index dens un fichier et sa relecture en début de session de travail évitent la perte de temps de sa constitution. La sauvegarde de la table d'index doit être faite à *chaque création* de clé et non pas seulement en fin de session.

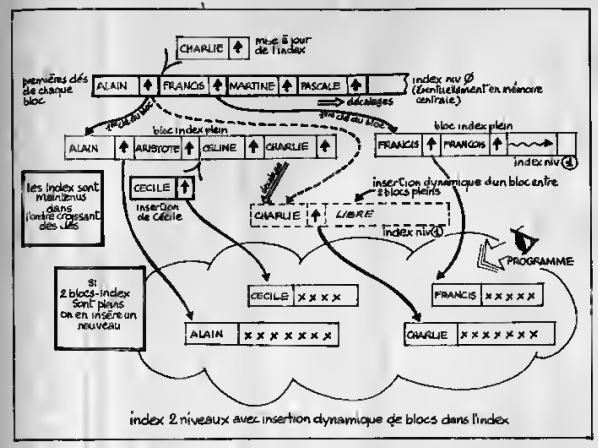
En effet, sans cette sauvegerde, si une session était interrompue, l'index sur disque ne sereit plus à jour pour les sessions suivantes.

Si l'on a choisi d'avoir cette table en ordre croissant afin d'y faire une recherche plus rapide par dichotomie, on doit bien sûr, lors d'une création de clé, insérer celle-ci dans la teble après avoir décalé toutes les clés en aval de l'insertion.

```
10 REM          ****************************
20 REM ACCES INDEXE AVEC TABLE D INDEX EN MEMOIRE CENTRALE
30 REM          CONSTITUEE EN DEBUT DE SESSION
40 REM          ****************************
50 REM          AUTEUR J. BOISGONTIER
60 REM   COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L AUTEUR
70 REM          ****************************
80 OPEN '#',1,"DATA"
90 FIELD #1, 10 AS N$, 8 AS PRS, 20 AS TLS, 4 AS NTS    'DEFINITION DES ZONES
100 DIM NOMS(50),INDEX(50)
110 GOSUB 1000                                          'CONSTITUTION DE LA TABLE D'INDEX
120 PRINT INPUT " MODE (CREAT,CONS,MODI,AR,NXT,LIST,FIN) ";MODE$
130 IF MODE$="CREAT"   THEN GOSUB 2000
140 IF MODE$="CONS"    THEN GOSUB 3000
150 IF MODE$="MODI"    THEN GOSUB 4000
160 IF MODE$="AR"      THEN GOSUB 5000
170 IF MODE$="NXT"     THEN GOSUB 6000
180 IF MODE$="LIST"    THEN GOSUB 7000
190 IF MODE$="FIN"     THEN 210
200 GOTO 120
210 CLOSE #1
220 END
1000 REM ---------------------------------------------------------
1010 REM          CONSTITUTION DE L'INDEX
1020 J=1
1030 FOR I=1 TO LOF(1)
1040     GET #1,I
1050     IF ASC(N$)=0 THEN 1070          'ENREGISTREMENT VIDE ?
1060     NOMS(J)=N$:INDEX(J)=I:J=J+1
1070 NEXT I
1080 RETURN
2000 REM ---------------------------------------------------------
2010 PRINT "CREAT : A REALISER" RETURN
3000 REM ---------------------------------------------------------
3010 PRINT "CONS : A REALISER" RETURN
4000 REM ---------------------------------------------------------
4010 PRINT"MODI  A REALISER" RETURN
5000 REM ---------------------------------------------------------
5010 PRINT"AR : A REALISER" RETURN
6000 REM ---------------------------------------------------------
6010 REM SOUS-PROGRAMME DE RECHERCHE INDEXEE
6020 X$="";INPUT "NOM CHERCHE ";X$:IF X$="" THEN 6120
6030 L=LEN(X$):TR=0
6040 FOR I=1 TO 200
6050     IF X$=LEFT$(NOMS(I),L) THEN TR=1:NE=I:I=200:GOTO 6070
6060     IF NOMS(I)="" THEN NE=-1:I=200
6070 NEXT I
6080 IF TR=0 THEN PRINT:PRINT"N'EXISTE PAS":PRINT:GOTO 6040
6090 GET #1,INDEX(NE)   'LECTURE DE L'ENREGISTREMENT POINTE PAR LA TABLE D'INDEX
6100 PRINT PRN$ NB_PN3,TL$
6110 GOTO 6020
6120 RETURN
7000 REM ---------------------------------------------------------
7010 PRINT "LIST : A REALISER" RETURN
7020 REM ---------------------------------------------------------
7030 'SUR ERR=90
8010 ' PENDER A PAIRE CoPAR 3000
8020 '  REMPLACER LOF(1) PAR LOF(1)+1
```







index 2 niveaux avec insertion dynamique de blocs dans l'index

Index continu sur disque en ordre croissant

La présence en mémoire centrale d'une table d'index de taille importante peut être évitée de la façon suivante : les clés sont rangées en ordre croissant dans un fichier index et une table d'index de taille réduite, en mémoire centrale, contient la première clé de chaque enregistrement du fichier index ; on sait ainsi, lorsqu'on recherche une clé, sur quel enregistrement du fichier index elle est située.

Après avoir retrouvé la clé dans le fichier index, on accède à l'enregistrement associé à cette clé. Deux accès disques ont donc été nécessaires pour accéder à l'enregistrement cherché.

De façon à maintenir le fichier index en ordre croissant, chaque nouvelle clé est insérée directement à sa place après avoir décelé toutes les clés en aval de l'insertion. Aussi le temps nécessaire pour un ajout de clé peut-il être long lorsque le nombre de clés devient important (quelques secondes). Si on considère que les ajouts de clés sont généralement beaucoup plus rares que les consultations, ceci est acceptable.

Notons que le temps d'insertion des clés sera d'autant plus court que les enregistrements du fichier index seront plus longs (mais on perdra alors de la place mémoire).

Index 2 niveaux

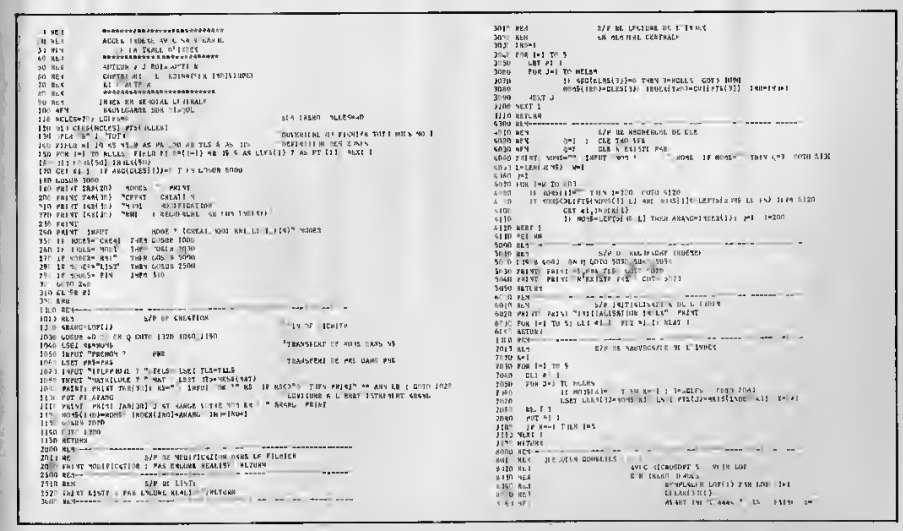
Plutôt que d'insérer des clés dans un fichier index continu, on peut allouer les enregistrements de cet index dynamiquement. A la création de l'index n'existe qu'un index de niveau 0. Ce niveau d'index peut être éventuellement situé en mémoire centrale.

Un index de niveau 1, constitué dynamiquement au fur et à mesure des ajouts de clés, doit être vu comme un index continu en ordre croissant.

Lorsqu'une insertion est faite dans un bloc (enregistrement) plein suivi d'un autre bloc plein, on alloue un nouveau bloc qui reçoit la dernière clé du premier bloc plein. La « continuité » de l'index 1 est assurée grâce à l'index 0. Tout se passe comme si nous avions inséré dans un index continu des *trous* qui seront utilisés lors d'insertions ultérieures de clés.

Ainsi il n'y a de décalages de clés qu'entre 2 blocs tout au plus.
Pour des enregistrements de 256 caractères, l'index pourra contenir jusqu'à 40 × 40 = 1 600 clés.

Jacques Boisgontier



RECHERCHE DICHOTOMIQUE DANS UNE TABLE

Le sous-programme de recherche dichotomique figure aux lignes 300 à 399 comprises, en remplaçement du sous-programme de recherche séquentielle qui figurait dans la fiche de la page 114

Les lignes 10 à 299 sont identiques à celles de la fiche de la page 114, à l'exception de la ligne 35, nécessaire du fait qu'il faut trier la table avant de pouvoir commencer la recherche.

Le fonctionnement du sous-programme est simple : on part d'une table triée (d'où la ligne 35) et l'on compare l'élément A$ que l'on cherche avec le plus petit et le plus grand élément de la table

— A$ est égal au plus petit élément T$(0), on a donc trouvé A$ dans la table à la position 0.

— A$ est égal au plus grand élément T$(N9) on a trouvé A$ dans la table à la position N9.

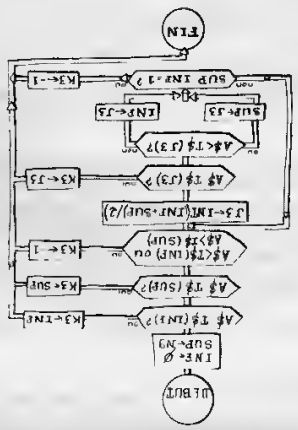
— A$ est plus petit que le plus petit élément T$(0) ou plus grand que le plus grand élément T$(N9) : on sait donc que A$ ne peut être dans la table

— dans tous les autres cas, A$ est susceptible d'être dans la table, quelque part entre INF (= 0) et SUP (= N9)

On procède alors par *dichotomie* (= division par deux), c'est-à-dire qu'on va partager l'intervalle INF/SUP en deux et regarder si A$ se trouve au milieu, ou dans l'une des deux moitiés. On travaille alors sur cette moitié, et on recommence cette division jusqu'à ce que

— soit l'on ait trouvé A$ dans la table à une position J3 ;

— soit SUP = INF + 1, et A$ n'est ni T$(INF), ni T$(SUP) comme A$ devrait se trouver « entre les deux », ce qui est impossible puisqu'on n'utilise que des valeurs entières, c'est donc que A$ n'est pas dans la table !




```
10 REM TRI D'UNE TABLE EN MEMOIRE CENTRALE
20 REM COPYRIGHT L'ORDINATEUR INDIVIDUEL
30 GOSUB100:REM INITIALISATIONS
35 PRINT"TRI DE LA TABLE":GOSUB400:PRINT"VOILA"
...
100 REM INITIALISATIONS
110 READN:REM NOMBRE DE MOTS DANS LA LISTE (MOINS UN)
120 DATA9
130 DIMT$(N)
...
300 REM RECHERCHE DICHOTOMIQUE DANS LA TABLE T$(N)
305 REM ** ATTENTION,O LA TABLE DOIT ETRE TRIEE !!
310 INF=0:SUP=N9
320 IF A$=T$(INF) THEN K3=INF:GOTO390
325 IF A$=T$(SUP) THEN K3=SUP:GOTO390
330 IF A$<T$(INF) OR A$>T$(SUP) THEN K3=-1:GOTO390
340 J3=INT((INF+SUP)/2)
345 IF A$=T$(J3) THEN K3=J3:GOTO390
350 IF A$<T$(J3) THEN SUP=J3:GOTO380
355 INF=J3
360 IF SUP=INF+1 THEN 340
365 K3=-1:REM IL DEVRAIT EXIST PAS GOTO390
390 IF K3=-1 THEN PRINT"CE MOT N'EXISTE PAS":GOTO399
395 PRINT"CE MOT EST LE MOT NO.",K3," DE VOTRE LISTE"
399 RETURN
...
```


RECHERCHE SEQUENTIELLE RAPIDE

En BASIC interprété, il est possible d'accélérer notablement une recherche séquentielle. Pour ce faire, il vous suffit d'utiliser une table.

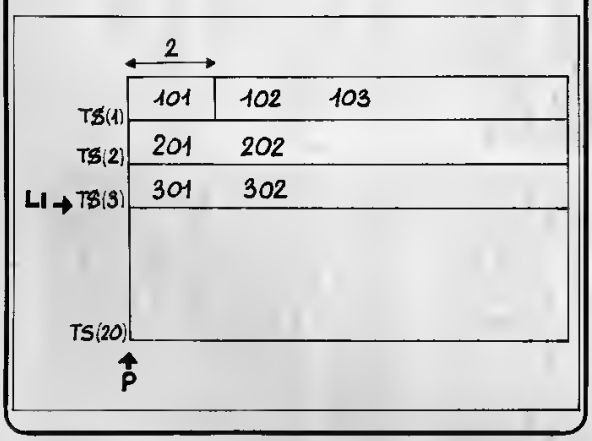
Prenons le cas d'une table de 2 000 nombres. On peut la définir comme une table de 20 chaînes de 100 X 2 caractères. Chaque couple de caractères représente alors un nombre compacté par la fonction MKI $.

En sens inverse, la recherche se fera alors à l'aide de la fonction INSTR.

Il est possible de sauvegarder ces chaînes dans des fichiers : le tampon est dans ce cas défini par :

```
FIELD # 1,200 AS I1 $
FOR I = 1 TO 20
LSET I1 $ = LI $ (I)
PUT # 1, I
NEXT I
```

L'organisation de votre table est alors représentée par la figure ci-dessous.



Remarque sur le programme : la ligne 340 correspond aux positions impaires seulement.

Liste du programme

```
150 CLEAR(3000)
160 DIM T$(20)
170 '
180 '--------------- Generation pour essai
190 FOR LI=1 TO 20
200   FOR I=1 TO 100
210     T$(LI)=T$(LI)+MKI$(LI*100+I)
220   NEXT I
230 NEXT LI
240 '-------------- Recherche
250 '
260 INPUT "Quel nombre cherchez vous? ";X
270 X$=MKI$(X)
280 '
290 FOR LI=1 TO 20                  ' 20 chaines
300  DR=1                           ' Debut de recherche
310 '
320  P=INSTR(DR,T$(LI),X$)
330  IF P=0 THEN 360
340  IF (P MOD 2)=0 THEN DR=P+1:GOTO 320
350  PRINT "Position de X:";P;LI;CVI(MID$(T$(LI),P,2))
360 NEXT LI
370 GOTO 260
```

Exemple d'exécution

```
run

Quel nombre cherchez vous? 1207
Position de X: 13 12 1207
```

TRI D'UNE TABLE PAR INSERTION DICHOTOMIQUE

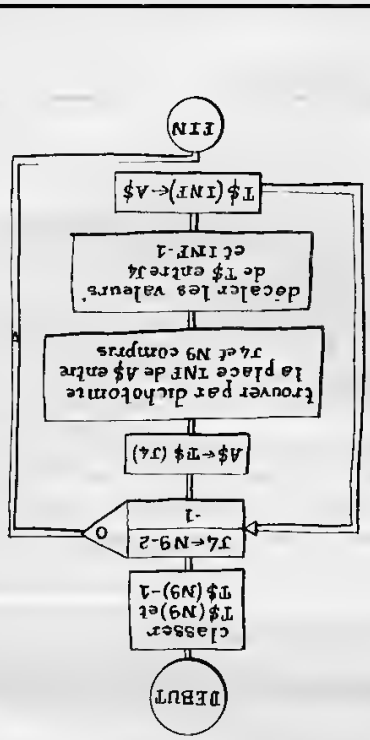
Ce sous-programme trie un tableau de N 9 + 1 chaînes de caractères si-tuées dans T$ (N9)

L'algorithme présenté ici est une méthode d'insertion Après quelques préliminaires (instructions 410 à 430) on va insérer l'élément actuellement situé en J4 dans la suite *déjà triée* des éléments J4 + 1 à N9

La place exacte de l'insertion est déterminée par une recherche dichotomique (cf fiche n° 10), lignes 435 à 455 L'insertion se fait alors (lignes 470 et 475), et le cycle recommence tant que toute la table n'est pas triée

Notez que, contrairement à ce qui se passe dans la fiche de la page 114 on ne compare pas directement A$ et T$ (N9) : ceci oblige à mettre à la ligne 440 : SUP = N9 + 1.

```
400 REM TRI DE LA TABLE T$(N9) DANS L'ORDRE CROISSANT
          (METHODE PAR INSERTION DICHOTOMIQUE)
410 IF N9+1=1 THEN 499
420 NI=N9-1:IF T$(N9)<T$(NI) THEN
          A$=T$(N9):T$(N9)=T$(NI):T$(NI)=A$
425 IF N9+1=2 THEN 499
430 FOR J4=N9-2 TO 0 STEP-1
435 A$=T$(J4):IF A$<=T$(J4+1) THEN 480
440 INF=J4:SUP=N9+1
445 K4=INT((INF+SUP)/2):
          IF A$<T$(K4) THEN SUP=K4:GOTO 455
450 INF=K4
455 IF SUP-INF>1 THEN 445
470 FOR K4=J4 TO INF-1:T$(K4)=T$(K4+1):NEXT K4
475 T$(INF)=A$
480 NEXT J4
499 RETURN
```



Cette fiche contient trois sous-programmes de manipulation de chaînes de caractères.

Dans de nombreuses utilisations, il est nécessaire de tester ou de modifier des chaînes de caractères
La présente fiche contient trois sous-programmes

En 9500, un sous-programme cherchant la position d'un caractère donné (X9$) dans une chaîne donnée (ZZ$) Cette position est inscrite dans la variable X7 et elle vaut zéro si X9$ ne figure pas dans ZZ$
Certains systèmes possèdent une fonction équivalente, qui s'appelle sous des noms tels que INSTR INDEX ou POS
Ce sous-programme (ou un équivalent) est utilisé pour d'autres sous-programmes

En 9700-9745, un sous-programme de conversion de majuscules en minuscules Il sera utile aux utilisateurs de PSI ne pouvant afficher de minuscules (Apple II TRS 80) pour convertir des chaînes et par exemple faire du traitement de texte

En 9750-9795, un sous-programme inverse qui transforme en majuscules des caractères minuscules

Ces deux derniers sous-programmes sont identiques, à deux lignes près

La fonction MID$ joue son rôle habituel d'extraction d'une chaîne depuis une autre Ainsi, A$ = MID$ (B$, I, J) aurait pour effet de mettre dans A$ la sous-chaîne de B$ à partir du caractère numéro I de B$, et en prenant J caractères au maximum

Pour ces deux sous-programmes, on examine successivement chaque caractère de la chaîne Z$, ou plus exactement son code tel qu'il est fourni par la fonction ASC (CODE pour certains BASIC) Si ce code est compris entre celui du A et du Z majuscules (ou minuscules suivant le cas), on y rajoute de quoi le transformer en son correspondant minuscule (respectivement, majuscule) Il ne reste plus alors qu'à replacer le caractère jusqu'au caractère exclu, suivi du caractère modifié, suivi du reste de la chaîne On distingue en fait deux cas supplémentaires et particuliers lorsque le caractère est le premier ou le dernier de la chaîne

```
9500 REM  TROUVER POSITION CARACTERE X9$ DANS CHAINE ZZ$
9505 X9=0
9510 ZZ=LEN(ZZ$)
9515 IF ZZ=0 THEN 9545
9520 FOR X8=1 TO ZZ
9525   IF MID$(ZZ$,X8,1)=X9$ THEN X9=X8:X8=ZZ
9530 NEXT X8
9545 RETURN
```

```
9700 REM CONVERSION DE Z$ DE MAJUSCULES EN MINUSCULES
9705 AA=ASC("A"):AI=ASC("a"):ZA=ASC("Z"):X0=AI-AA
9710 X9=LEN(Z$)
9712 IF X9=0 THEN 9745
9714 FOR X8=1 TO X9
9718     X7=ASC(MID$(Z$,X8,1))
9720     IF X7 < AA  OR X7 > ZA THEN 9740
9722     X7$=CHR$(X7+X0)
9724     IF X8=1  THEN Z$=X7$+MID$(Z$,2,X9-1):GOTO 9740
9726     IF X8=X9 THEN Z$=MID$(Z$,1,X9-1)+X7$:GOTO 9740
9728     Z$=MID$(Z$,1,X8-1)+X7$+MID$(Z$,X8+1,X9-X8)
9740 NEXT X8
9745 RETURN

9750 REM CONVERSION MINUSCULES EN  MAJUSCULES
9755 AA=ASC("A"):AI=ASC("a"):ZI=ASC("z"):X0=AA-AI
9760 X9=LEN(Z$):IF X9=0 THEN 9795
9762 FOR X8=1 TO X9
9764     X7=ASC(MID$(Z$,X8,1))
9766     IF X7 < AI OR X7 > ZI THEN 9790
9768     X7$=CHR$(X7+X0)
9770     IF X8=1  THEN Z$=X7$+MID$(Z$,2,X9-1):GOTO 9790
9772     IF X8=X9 THEN Z$=MID$(Z$,1,X9-1)+X7$:GOTO 9790
9774     Z$=MID$(Z$,1,X8-1)+X7$+MID$(Z$,X8+1,X9-X8)
9790 NEXT X8
9795 RETURN
```

Première parution : L OI n° 12

```
10 REM EXEMPLE D'UTILISATION
15 REM *** TEST DE NUMERICITE D'UNE ZONE ***
20 CLS
25 Z9$="0123456789"
30 INPUT"ZONE A VERIFIER ";Z1$
40 GOSUB 9550
50 IF X6=0 THEN PRINT"ZONE NUMERIQUE"
                ELSE PRINT"ZONE NON NUMERIQUE"
60 INPUT"AUTRE ZONE A VERIFIER ";X$
70 IF X$="O" THEN GOTO 30
80 END

9550 REM VERIFIER QUE TOUS CARACTERES DE Z1$ FIGURENT DANS Z9$
9555 X6=0:Z1=LEN(Z1$):IF Z1=0 THEN 9595    :REM UTILISE 9500
9560 Z2$=Z9$
9565 FOR X7=1 TO Z1
9570    X9$=MID$(Z1$,X7,1)
9580    GOSUB 9500
9585    IF X9=0 THEN X6=X7:X7=Z1: REM X9$ PAS DANS Z2$
9590 NEXT X7
9595 RETURN

9600 REM ELIMINER DE Z1$ TOUT CARACTERE NE FIGURANT PAS DANS Z9$
9610 GOSUB 9550                             :REM UTILISE 9550
9615 IF X6=0 THEN 9645
9620 IF X6=1 THEN Z1$=MID$(Z1$,2,LEN(Z1$)-1)      :GOTO 9640
9625 IF X6=LEN(Z1$) THEN Z1$=MID$(Z1$,1,LEN(Z1$)-1):GOTO 9640
9630 Z1$=MID$(Z1$,1,X6-1)+MID$(Z1$,X6+1,LEN(Z1$)-X6)
9640 GOTO 9610
9645 RETURN

9650 REM SUPPRIMER DE Z2$ TOUS CARACTERES FIGURANT DANS Z9$
9655 Z1$=Z2$:Z2$=""
9660 GOSUB 9550                             :REM UTILISE 9550
9665 IF X6 <> 0 THEN Z2$=Z2$+MID$(Z1$,X6,1):
                     Z1$=MID$(Z1$,X6+1,LEN(Z1$)-X6):
                     GOTO 9660
9680 RETURN
```

Fiche pratique — **CHAINES DE CARACTERES**

Cette fiche contient trois sous-programmes de manipulation de chaînes de caractères.

Dans de nombreuses situations, il est nécessaire de tester ou de modifier des chaînes de caractères.

La présente fiche contient trois sous programmes, le premier utilisant le sous programme 9500-9545 de la fiche 14 et utilisé par les deux autres. On utilisera donc ces quatre sous programmes conjointement.

En 9550, un sous-programme qui vérifie que tous les caractères de Z1$ figurent dans Z9$. Si oui, la variable X6 contient 0. Si non, la variable X6 contient le numéro du premier caractère de Z1$ ne figurant pas dans Z9$. Le programme principal donne (lignes 10 à 80) montre un exemple d'utilisation de ce sous-programme pour vérifier si le contenu d'une chaîne est bien numérique : c'est-à-dire ne contient que des caractères pris dans 0123456789.

En 9600, un sous programme qui permet d'éliminer de Z1$ les caractères ne figurant pas dans Z9$. On peut ainsi, par exemple épurer Z1$ de tous les caractères non numériques : c'est-à-dire utiliser pour Z9$ la chaîne 0123456789.

En 9650, un sous-programme qui élimine de Z2$ tous les caractères figurant dans Z9$. On peut par exemple supprimer ainsi tous les caractères point, blanc, virgule d'un texte.

Fiche pratique

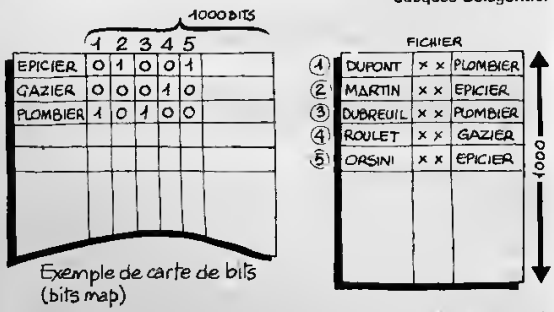
LISTE INVERSEE

A côté de l'accès direct et de l'accès séquentiel, existe l'accès indexé. Cette méthode (dont nous vons avons déjà parlé à la page 111) combine les avantages de l'accès séquentiel (peu de perte de place) et de l'accès direct (temps d'accès réduits). Toute personne travaillant sur des fichiers de données qu'elle désire utiliser à l'aide d'une ou plusieurs clés se doit, à terme, d'utiliser des techniques d'accès indexé. Si vous vous êtes constitué un fichier des personnes que vous invitez à dîner et des préférences et contre-indications gastronomiques de celles-ci, vous devrez, lors de la préparation d'un repas donné, voir si ce dernier est compatible avec les goûts culinaires de vos invités ou vous devez reconstituer un repas potentiel d'après ces mêmes goûts culinaires. Ceci n'est qu'un exemple d'index se rapportant partiellement à un fichier donné (en l'occurence l'index des invités du repas considéré par rapport à l'ensemble de vos connaissances).

Les systèmes de fichiers actuels ne permettent pas d'accéder aux enregistrements par leur contenu (mémoires associatives). Il faut, par exemple lorsque l'on recherche dans un fichier toutes les personnes exerçant une certaine profession, explorer celui-ci dans sa totalité en ne sélectionnant que les enregistrements qui correspondent à la profession cherchée.

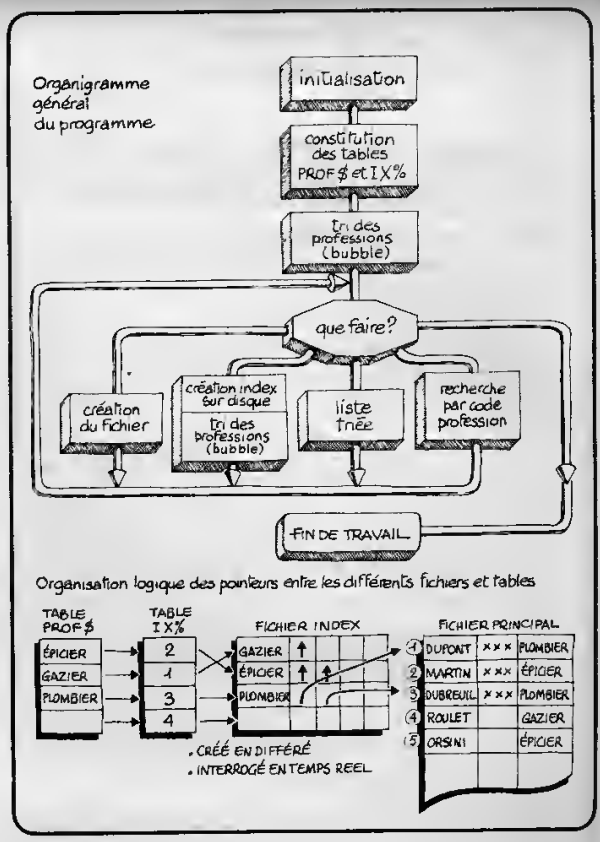
Ceci peut évidemment être long. Aussi est-on amené à créer un index contenant pour chaque profession des pointeurs vers toutes les personnes exerçant cette profession. Sur l'exemple, l'index est créé en différé, mais rien n'empêche de le mettre à jour à chaque ajout d'enregistrement dans le fichier des noms. Plusieurs index pour différents « critères » peuvent être gérés et exploités simultanément en « temps réel » ou en différe.

Jacques Boisgontier



Exemple de carte de bits (bits map)

Des « BITS MAP » pourraient remplacer ces systèmes de pointeurs : soit par exemple un fichier de 1 000 noms. On réserverait pour chaque profession 1 000 bits, chacun de ces bits indiquant si une personne exerce ou non cette profession.



```
3010 REM *** TRI DES PROFESSIONS
3020 IF J<K2 THEN 3100
3030 FOR I=1 TO IB-1                    :REM TRI PROFESSIONS
3040 FOR J=I+1 TO IB                    :REM TRI SYMBOLE
3050 IF PROF$(J)>=PROF$(I) THEN 3080
3060 IS=PROF$(I):PROF$(I)=PROF$(J):PROF$(J)=IS
3070 IZ=INX%(I):INX%(I)=INX%(J):INX%(J)=IZ
3080 NEXT J
3090 NEXT I
3100 RETURN
4000 REM----------------------------------------
4010 REM *** LISTE PAR PROFESSION SUR IMPRIMANTE
4020 PRINT:PRINT"Profession:";PROF$(I);"Nom";PRINT
4030 FOR J=1 TO LOF(2)                  :REM LISTE PAR PROFESSION
4040 GET #2,L$:IX$(COUN)
4050 IF ASC(PX$)=0 THEN 4120
4060 PRINT PX$;
4070 FOR I=1 TO 110
4080 IF ASC(PX$(I))=IX THEN 4110
4090 GOTO 4120
4100 NEXT I
4110 PRINT
4120 NEXT COUN
4130 RETURN
5000 REM----------------------------------------
5010 REM *** CONSTITUTION DES TABLES PROFS AT IXX
5020 IB=0
5030 FOR I=1 TO LOF(2)
5040 GET #2,L$
5050 IF ASC(PX$)=0 THEN 5070
5060 IB=IB+1 :PROF$(IB)=PX$:IXX(IB)=I
5070 NEXT I
5080 RETURN
6000 REM----------------------------------------
6010 REM *** INTERROGATION PAR PROFESSION
6020 PRINT:X$="":INPUT"PROFESSION ACCUEILLIE";X$
6030 IF X$="" THEN 6150
6040 I=0
6050 FOR IH=1 TO IB
6060 IF X$=PROF$(IV) THEN I=IV:IH=IB
6070 NEXT IH
6080 IF I=0 THEN PRINT"* CETTE PROFESSION N'EXISTE PAS":GOTO 6020
6090 FOR J=1 TO 110
6100 GET #2,L$(J):PRINT
6110 NEXT K
6120 PRINT:PRINT"Nombre";IB;"professions"
6130 PRINT K
6140 GOTO 6020
6150 RETURN
7000 REM---------- FIN ----------
```

```
1000 REM
1010 REM    LISTE INVERSE
1020 REM    OPERANT D'ORDRE ALPHA POUR ITEM
1030 REM    AFFICHE INDICE ITEM
1040 REM    ET INDICATEUR DE L'ATTRIB
1050 REM
1060 REM
1070 REM
1080 REM
1090 REM
1100 REM    CE PROGRAMME FAIRE A PARTIR D'UN FICHIER DE CONSTITUER
1110 REM
1120 REM    DES PROFESSIONS ACCORDANT TOUTES LES PERSONNES EXERCANT LA MEME
1130 REM    UNE INTERROGATION SUR TRIS NOMS, PUIS ENFIN PARTE A L'AIDE DE CET INDEX
1140 REM    DU DIFFUSE SUR DISQUE UN INDEX
1150 REM    PROFESSION
1160 REM    POUR ACCROUVER IMPRIMIRAIENT TOUTES LES PERSONNES AVANT DANS CONTRAIN
1170 REM    PROFESSION
1180 REM
1190 REM
1200 REM
1210 REM
1220 REM
1230 REM
1240 REM
1250 REM
1260 REM
1270 REM
1280 REM
1290 REM
1300 REM
1310 REM
1320 REM
1330 REM
1340 REM
1350 REM
1360 REM
1370 REM
```

```
9930 REM -- LECTURE D'UN NOMBRE POSITIF.
9805 :
9810 REM ACCEPTE LES CHIFFRES DE LA FORME :
9815 REM XX.(CR) , XXX.(CR) OU XX.XXX(C)
9820 REM LES X ETANT DES CHIFFRES DE 0 A 9
9825 REM ET LE 1ER MAXI 10E DE CHIFFRES
9830 REM ETANT DE 11.
9835 REM LES EVENTUELS ZEROS EN DEBUT DE VALEUR SONT
9840 REM PRIS EN COMPTE.
9845 :
9850 Z=0:REM VALEUR LUE.
9855 PT=-1:REM PRESENCE D'UN POINT.
9860 N$=ASC(":")
9865 A9=ASC("9")
9870 Z$="":REM CHAINE DU NOMBRE.
9875 K$="":REM CARACTERE LU.
9880 :
9885 PRINT K$;:IF LEN(Z$)>11 THEN 9935
9890 GOT K$:IF K$="" AND PT=-1 THEN 9930:REM UN POINT AU PLUS
9895 IF K$="." THEN Z$=Z$+K$:GOTO 9885
9900 :
9905 IF K$<"0" OR K$>A9 THEN Z$=Z$+K$:GOTO 9885
9910 IF <>20 OR Z$="" THEN 9930:REM ZO-CODE DE DEL.
9915 N$=LEFT$(Z$,1):IF N$="" THEN PT=0
9920 PRINT"!":A$;"!":IF LEN(Z$)=1 THEN Z$="":GOTO 9890
9925 Z$=MID$(Z$,1,LEN(Z$)-1):GOTO 9890
9930 IF <>13 OR LEN(Z$)=0 THEN 9990:REM 13-CODE DE RETURN.
9935 REM - ON A LA CHAINE COMPLETE.
9940 REM ON PEUT MAINTENANT UTILISER VAL
9945 REM SANS PROBLEMES.
9950 Z=VAL(Z$)
9955 PRINT:REM ON VA LA LIRE.
9960 RETURN
```


Fiche pratique

ENTREE D'UN NOMBRE POSITIF

Ce sous-programme en BASIC permet l'entrée d'une valeur numérique positive.

Ce sous-programme est une variante du sous-programme de la Fiche Pratique n° 22, une chaîne de caractères est lue au clavier, chaque caractère étant vérifié au fur et à mesure de sa saisie.

La traduction de la chaîne de caractères en valeur numérique est ici faite grâce à la fonction VAL, mais il serait assez facile de s'en passer.

Notez une différence importante avec le sous-programme d'entrée de chaîne : on ne doit autoriser qu'un seul point au plus, d'où un certain nombre de précautions prises grâce à la variable PT (lignes 9855, 9895, 9905 et 9915).

Les valeurs numériques acceptées ici comportent 11 caractères au plus (point compris), les zéros éventuellement tapés au début de la valeur comptant. Dans la limite des 11 caractères, toutes les formes comportant au plus un point décimal sont acceptées.

Description du programme

Le sous-programme fournit une chaîne Z$ contenant une valeur numérique, et la valeur Z correspondante. Il utilise les variables PT (présence/ absence d'un point), AZ (code ASCII du caractère zéro), A9 (code ASCII du caractère 9), R$ (dernier caractère lu) et R (code ASCII de ce caractère).

Fonctions utilisées : ASC, LEN, GET (à remplacer suivant le système), RIGHT$, MID$ et VAL.

Fonctionnement identique au programme de la fiche n° 22, avec les contraintes supplémentaires dues à la longueur maximum autorisée (ici 11, cf ligne 9885) et à la présence d'un point au plus.

Adaptation

Le point le plus important est sans doute le changement du nombre maximum autorisé de chiffres, afin de tenir compte du nombre de chiffres significatifs du BASIC utilisé.

PROGRAMME STANDARD DE MENU

Ce programme vous permettra facilement de donner une petite touche « professionnelle » à vos programmes, en présentant une façon normalisée d'utiliser ceux-ci.

Le programme est très simple et se passe quasiment de commentaires. Notez cependant qu'à la ligne 420 on utilise GOSUB afin d'alléger la présentation du programme, ce qui conduit à « tricher » pour le « sous-programme » 500 qui en fait arrête l'exécution du programme 1000 sert aux initialisations de l'ensemble du programme : tous les READ et DATA devraient s'y trouver pour limiter les problèmes.

Complétez ce programme suivant vos besoins, en tenant compte des suggestions et remarques ci-dessous.

Si vous voulez forcer l'écriture systématique des commandes autorisées avant chaque demande d'une commande, placez en 1BO une instruction GOSUB 3000.

Si vous souhaitez ajouter ou supprimer des commandes, il vous faut modifier les lignes 410, 420, 1010, 1040, et les lignes 4020 à 4100. Dans le sous-programme 4000 les commandes sont données dans l'ordre alphabétique, mais ce n'est pas obligatoire.

Il vous faut bien entendu compléter les sous-programmes 5000, 6000 et 7000. Ce dernier comportera en fait le texte du programme que vous voulez « habiller » moins la partie initialisation que vous aurez mise au sous programme 1000.

La liste du programme

```
100 REM PROGRAMME STANDARD DE MENU
110 REM CONVIENT A PRESQUE TOUTES
120 REM APPLICATIONS.
130
140 GOTO 1000          REM INITIALISATIONS
150
160 L=-8 K=0           REM NOUVELLE COMMANDE
170
180 REM LECTURE D'UNE COMMANDE ET VALIDATION
190 K$="":PRINT C ;INPUT K$ ;IF K$="" THEN 190
200 CH=ASC(K$+A 3)     REM CH VA PRENDRE LES 3 PREMIERS CARACTERES
210 FOR J=1 TO NC      REM POUR CHACUNE DE LA COMMANDE
220 IF CH=CS(J) THEN J=NC      REM TROUVE
230 NEXT J
240 IF L>0 GOSUB 500
250
260 REM COMMANDE ILLEGALE
270 PRINT "?";K$
280 REM CONTROLE D'ERREUR SUR CETTE COMMANDE
290 IF L<3 THEN PRINT "ATTENDEZ !" ;REM UNE ERREUR DES COMMANDES AUTORISEES K=0
300 GOTO 100
310
400 REM EXECUTION DUNE COMMANDE LEGALE
410 GOSUB 500,2000,3000,4000,5000,6000,7000
420 ON L GOSUB 500,
430 GOTO 160
440
500 REM FIN
510 REM EL "SOUS PROGRAMME" ARRETE LA FAIT L'EXECUTION
520 REM RETOURNE A PARTIR DES PROGRAMMES
530 PRINT:PRINT
540 END
550
```


La liste du programme (suite)

```
990 REM ---------------- SOUS-PROGRAMMES ----------------
1000 REM                INITIALISATIONS
1010 RC=8               REM NOMBRE DE COMMANDES
1020 DIM CS(NC)
1030 FOR J=1 TO NC READ CS(J);NEXT J
1040 DATA FIN,VLN,*,SOS,AIN,INS,POS,SAR
1050
1060 C=0                REM NOMBRE DE COMMANDES EXECUTEES
1070
1080 RETURN
1090
1100 REM                VERSION
2000 REM                ----
2010 PRINT "VERSION I - 1/6/80 PAR L'OI
2020 RETURN
2030 REM
2040
2050 REM                ----
2060 FOR J=1 TO NC      ----
2070 PRINT CS(J);" ";   ----
3000 REM
3010 FOR J=1 TO NC      ----
3020 PRINT CS(J);" ";   ----
3030 NEXT J
3040 PRINT
3050 RETURN
3060 REM
3070
3080
3090 REM                ----
4000 REM                ----
4010 PRINT:PRINT"LISTE DES COMMANDES AUTORISEES :" ;PRINT
4020 PRINT AIDE         RAPPELE CETTE LISTE
4030 PRINT"EXECUTION :  LANCE L'EXECUTION DU PROGRAMME.
4040 PRINT FIN          TERMINE L'EXECUTION DU PROGRAMME
4050 PRINT INSTRUCTIONS DONNE LES INSTRUCTIONS POUR L'UTILISATION
4060 PRINT              DU PROGRAMME
4070 PRINT POSSIBILITES EXPLIQUE LES POSSIBILITES DU PROGRAMME
4080 PRINT"SOS (OU ?)   APPELLE UNE LISTE BREVE DES COMMANDES"
4090 PRINT VERSION      DONNE LA DATE DE DERNIERE MISE A JOUR DU
4100 PRINT              PROGRAMME
4110 PRINT: PRINT"SEULS LES TROIS PREMIERS CARACTERES D'UNE COMMANDE SONT
4120 PRINT"PRIS EN COMPTE"
4130 RETURN
4140 REM
4990
5000 REM                ----
5010 PRINT "POUR UTILISER LE PROGRAMME ,
5020 PRINT TAPEZ 'EXE', LE PROGRAMME VOUS DEMANDERA ALORS ..."
5030 PRINT"PUIS.... ENTER (A COMPLETER)"
5040 RETURN
5980 REM                ----
5990 :
6000 REM                ----
6010 PRINT"CE PROGRAMME PERMET D'OBTENIR ... A PARTIR DE ...."
6020 PRINT"CT JE ... (A COMPLETER)
6030 RETURN
6980 REM                ----
6990
7000 REM                ----
7010 PRINT "EXECUTION
7020 RETURN
7030 REM
```

Lorsque l'on entre un programme au clavier il est toujours désagréable et astreignant de le présenter tout de suite correctement. Alors pourquoi ne pas écrire un programme qui effectue automatiquement ce travail ? Et en primo, il vous indiquera la présence de boucles FOR-NEXT non appariées car il ne pourra pas les cadrer normalement dans ce cas.

Ce programme se propose deux buts
réaliser le decalage des boucles FOR NEXT
cadrer les commentaires en colonne 60.
Ceci permet une meilleure lisibilité des programmes et facilite d'autant leur mise au point.

La version choisie est la version Microsoft comportant des espaces séparateurs. Dans le cas du TRS-80 il n'y a pas de separateur et on n'accepte donc pas les noms de variables comportant les mots FOR et NEXT. On pourra donc simplifier le programme dans ce cas.

Le programme que l'on désire presenter recadré devra prealablement avoir ete sauvegarde sous forme ASCII — c'est à-dire non préinterprétée. La commande nécessaire est du type SAVE « XXX » A

Décalage des boucles FOR-NEXT

Le programme considere trois niveaux successifs de boucles — du type FOR I — FOR J - FOR K - NEXT K - NEXT J - NEXT I

Dans le cas où il repere qu'un NEXT ne correspond pas au FOR précédent il l'indique par le message « FOR NEXT PAS APPARIES EN » numéro de ligne

Lorsqu'il se trouve en presence d'un FOR sans NEXT — cas d'un FOR NEXT sur la même ligne — il garde le cadrage précédent

La presence de deux FOR sur la même ligne amene un cadrage décalé (voir l'exemple d'execution) de droite à gauche il est possible d'éviter ceci en provoquant des sauts de ligne lors de la rencontre de plus d'un ordre sur une même ligne

Cadrage des commentaires

Les commentaires sont tous alignés sur la 60e colonne de la liste — variable CC — Sont pris en compte comme commentaires tout ce qui suit les caractères ' REM et ' REM

Pour l'ensemble du programme on considere que les enregistrements lus font 255 octets de long

Exemple d'exécution

```
10 REM Ce n'est un essai de decalage des boucles FOR-NEXT
20 REM
30 FOR VUM=1 TO 10
40 PRINT FOR
50 FOR K=1 TO 200: PRINT K: NEXT K:        REM FOR SANS NEXT
60 PRINT FOR
70 FOR TRUC=1 TO 20
80 FOR J=1 TO 5
90 PRINT J
100 PRINT TRUC
110 NEXT TRUC
120 NEXT VUM
130 REM-------------------------------------------------------------
140 REM  XXXXX 'YYY'                    REM   'XXXXXX'
150 REM
160 PRINT "ESSAI"                       REM XXXXX
170 PRINT ESSAI"                        REM BSAT BOUCLE FOR
180 FOR K=1 TO 20                       REM COMMENTAIRE
190 REM
200 PRINT "REM"
210 PRINT 1
220 FOR J=1 TO 5                        REM NEXT   XXXXXXX
230 PRINT FOR
240 NEXT I,J
250 NEXT K
260 REM
270 REM-------------------------------------------------------------
280 PRINT  FOR I=1 TO 10 PRINT " FOR   FOR Z=1 TO 3
290 PRINT P
300 PRINT P NEXT P
310 PRINT 1
320 NEXT 1
330 REM-------------------------------------------------------------
340 FOR I=1 TO 10
350 IF I=2 THEN I=9 NEXT I GOTO YX      REM FOR-NEXT PAS APPARIES
360 FOR Z=1 TO 10
370 NEXT 1
380 REM-------------------------------------------------------------
390 FOR I=1 TO 10                       REM Recadrage automatique
400 PRINT 1
410 PRINT 1
420 PRINT "COUCOU,XXX" NEXT 1
430 REM-------------------------------------------------------------
440 REM
450 FOR M=1 TO 3
460 LNEXT=2 NEXTL=5
470 FOR I=1 TO 10 FOR J=1 TO 5:         REM 2 FOR sur la meme ligne
480 PRINT LNEXT,NEXTL
490 NEXT J: NEXT I
500 NEXT K
```


Ces programmes en BASIC permettent de réaliser soit l'intersection, soit le fusion de deux tables triées.

On a souvent besoin, étant données deux tablas triées A et B d'en axtraire une tabla triée C où est réalisée soit la fusion, soit l'intersection des deux tables initiales.

La logique d'incrémentation des pointeurs da tables est très simple.

Notez l'utilisation, comme indicateur de fin da liste ou de fichier du chiffre 9999. L'idéa est d'employar un indicateur qui n'aura jamais à apparaître dans la liste . un nombra non binaire dans une liste binaire, un nombre négatif dans une liste d'éléments tous positifs, une chaîne de caractères particuliers (des = ou des *) dans une liste alphanumérique.

On peut désirer connaître, dans la cas de la fusion, la provenance de chacun des éléments de la liste résultante · liste A, liste B ou les deux Ceci est réalisé en créant une liste C$ qui sera éditée en même temps que la liste C.

On peut ainsi détectar les répétitions entre les deux listes origines.

Une autre façon est de faire exécuter la routine d'intarsection des deux listes origine. On y remarque également la logique d'avance dans les tables qui évite de comparer successivement chaque élément da A avac tous les éléments de B en utilisant le fait qu'elles sont triées.

Une amélioration (simple !!!) consista à ajoutar la même indicateur de fin de table (9999) pour C.

```
10 REM*****************************************************************
20 REM          INTERSECTION DE 2 TABLES TRIEES
30 REM*****************************************************************
40 REM          COPYRIGHT : L'ORDINATEUR INDIVIDUEL
50 REM                      ET L'AUTEUR
60 REM*****************************************************************
70 REM  EXEMPLE :
80 REM
90 REM          TABLE A    ET    TABLE B  --->  TABLE C
100 REM         -------           -------        -------
110 REM         :  2  :           :  1  :        :B  1  :
120 REM         :  4  :           :  2  :        :AB 2  :
130 REM         :  5  :           :  3  :        :B  3  :
140 REM         :  7  :           :  4  :        :A  4  :
150 REM         : ... :           : 5   :        :A  5  :
160 REM         : 9999 :          : ... :        :A  8  :
170 REM                           : 9999 :       : ...  :
180 REM                                          :      :
190 REM*****************************************************************
```



```
200 REM SEQUENCE D'ESSAI
210 DIM A(10),B(10),C$(20),C(20)
220 FOR I=1 TO 9
230     A(I)=2*I : B(I)=I
240 NEXT I
250 A(10)=9999 : B(10)=9999
260 GOSUB 330
270 PRINT "TABLE A","TABLE B","TABLE C"
280 FOR I=1 TO 10
290     PRINT TAB(2)" ",TAB(4) A(I),TAB(10) " ",TAB(15)" ",TAB(17)B(I)
,TAB(23) " ",TAB(30)" ",TAB(31)C$(I),TAB(35)C(I),TAB(40)
300 NEXT I
310 END
320 *****************************************************************
330 REM                 PROGRAMME
340 L=1 :J=1 : K=1
350 REM BOUCLE DE MISE A JOUR
360 IF A(L)<B(J)        THEN    C$(K)="A ":C(K)=A(L)        :L=L+1
    ELSE IF A(L)>B(J)   THEN    C$(K)="B ":C(K)=B(J)        :J=J+1
    ELSE                        C$(K)="AB":C(K)=A(L)        :L=L+1 :J=J+1
370 K=K+1
380 REM TEST AVANT RETOUR EVENTUEL
390 IF A(L)<9999 OR B(J)<9999 THEN GOTO 350
400 RETURN
```

```
10 REM*****************************************************************
20 REM          FUSION DE 2 TABLES TRIEES
30 REM*****************************************************************
40 REM          AUTEUR : JACQUES BOISGONTIER
50 REM          COPYRIGHT : L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
60 REM*****************************************************************
70 REM  EXEMPLE .
80 REM
90 REM          TABLE A    ET    TABLE B  --->  TABLE C
100 REM         -------           -------        -------
110 REM         :  2  :           :  1  :        :B  1  :
120 REM         :  4  :           :  2  :        :AB 2  :
130 REM         :  6  :           :  3  :        :B  3  :
140 REM         :  8  :           :  9  :        :A  4  :
150 REM         : ... :           : 10  :        :A  6  :
160 REM         : 9999:           : ... :        :A  8  :
170 REM                           : 9999:        : ...  :
180 REM                                          :      :
190 REM*****************************************************************
200 REM SEQUENCE D'ESSAI
210 DIM A(10),B(10),C$(20),C(20)
220 FOR I=1 TO 9
230     A(I)=2*I : B(I)=I
240 NEXT I
250 A(10)=9999 : B(10)=9999
260 GOSUB 330
270 PRINT "TABLE A","TABLE B","TABLE C"
280 FOR I=1 TO 10
290     PRINT TAB(2)" ",TAB(4) A(I),TAB(10) " ",TAB(15)" ",TAB(17)B(I)
,TAB(23) " ",TAB(30)" ",TAB(31)C$(I),TAB(35)C(I),TAB(40)
300 NEXT I
310 END
320 *****************************************************************
330 REM                 PROGRAMME
340 L=1 :J=1 : K=1
350 REM BOUCLE DE MISE A JOUR
360 IF A(L)<B(J)        THEN    C$(K)="A ":C(K)=A(L)        :L=L+1
    ELSE IF A(L)>B(J)   THEN    C$(K)="B ":C(K)=B(J)        :J=J+1
    ELSE                        C$(K)="AB":C(K)=A(L)        :L=L+1 :J=J+1
370 K=K+1
380 REM TEST AVANT RETOUR EVENTUEL
390 IF A(L)<9999 OR B(J)<9999 THEN GOTO 350
400 RETURN
```

Liste du programme

```
10 REM *  CAMILLE DUPRAZ *
20 REM *  AUTOMOBILISTES ROUSSEAU *
30 REM *  CONTRIBUTION (NOVEMBRE) IL L'ANALYSE *
40 REM ******************************************
50 REM  Ce programme permet d'utiliser un programme en cadence les boucles FOR
60 REM  et les conditions.
70 REM
80 REM  ENCODAGE DE DIVERSES BOUCLES 'FOR' (FOR ET NEXT ARTIFICES)
90 REM  (Si FOR-NEXT pas applique recherche successive sur une
100 REM  boucle FOR exterieure)
110 REM
120 REM  CODAGE des COMMENTAIRES en COLONNE 60
130 REM
140 REM
150 REM  VERSION MICROSOFT AVEC PSEUDONYMES SEPARATEURS(A ADAPTER POUR TRS 80)
160 REM
170 REM
180 REM  Début du programme
190 REM  Ligne dans le tableau
200 REM  LIGNES   Ligne dans le tableau
210 REM  LIMITS   Ligne dans les commentaires
220 REM
230 NMAX=5
240 LIGNES=
250 PRINT "NOM DU PROGRAMME";"PGMS
260 OPEN "I",#1,PGMS
270 INPUT #1,LIGNES
280 IF X0E(1)=-1 THEN STOP
290 REM
300 REM  Boucle de recherche en ASCII
310 REM
320 REM  Ligne de commentaires
330 REM  LIMITS   Ligne dans le tableau
340 REM  LIGNES   Ligne dans le tableau
350 REM
360 REM  Separation programme de commentaires
370 REM
380 REM  ---- ------ ' ---------- ' XXXXXX  Analyse commentaires
390 REM  P1  P2  P3  P4
400 REM  |   |   |   |
410 REM
420 CC=0
430 REM  Caractère commentaire
440 REM
450 LIGNES=" "+LIGNES
460 REM  Lecture d'un programme
470 E1=0
480 E2=0
490 E3=0
500 E4=0
510 REM
520 E1=INSTR(E1+1,LIGNES,CHR$(34))
530 IF E1>0 THEN E2=INSTR(E1+1,LIGNES,CHR$(34))
540 IF E2>0 THEN E3=INSTR(E2+1,LIGNES,CHR$(34))
550 ELSE E2=0
560 REM Insère différence à l'intérieur de chaque chaîne de caractères
(ici LIGNES) ce donne ici elle exists, le dieses a laquelle elle se trouve
570 REM (donc le code ASCII est égal à 34)
580 REM CHR$(34) est équivalent au signe " (donc le code ASCII est égal à 34)
590 IF LIGNES="" THEN GOTO 400
600 REM
610 LIGNES=LIGNES+" "
620 E4=INSTR(E3+1,LIGNES,CHR$(34))
630 IF E4>0 THEN E3=INSTR(E4+1,LIGNES,CHR$(34))
640 CC(1)=INSTR(E3+1,LIGNES," ")
650 CC(2)=INSTR(E4+1,LIGNES," ")
660 REM
670 REM
680 REM  Recherche du plus petit
690 IF E(1)>0 THEN IF E(I)<E THEN E=E(I)  RE=I
700 NEXT I
710 FOR I=1 TO 3
720 IF E=255
730 REM
740 REM
750 IF E=255 THEN E=0
760 REM
770 IF E<0 THEN IF E2<1 AND E2<2 THEN E=E2  GOTO 620
780 IF E<0 THEN LTMAITS=LEFTS(LIGNES,E-1)
790 REM
800 REM
810 REM  Recherche FOR et NEXT
820 REM  ---- FOR ---------- FOR "------ FOR T=1 TO 10  Analyse d'une ligne
830 REM  P1  P2  P3  P4
840 REM  |   |   |   |
850 REM
860 REM
870 REM
880 REM
890 REM
900 REM
910 REM
920 REM
930 REM
940 REM
950 REM
960 REM  Liste de la chaîne à analyser
970 REM  P1 : Premier quote
980 REM  P2 : Deuxième quote
990 REM  P3 : Présence de FOR dans la ligne
1000 REM  P4 : Présence de NEXT
1010 REM
1020 REM
1030 REM
1040 REM
1050 REM
1060 REM
1070 REM
1080 REM
1090 REM
1100 REM
1110 REM
1120 REM
1130 REM
1140 REM
1150 REM
1160 REM
1170 REM
1180 REM
1190 REM
1200 REM
1210 REM  Traitement de NEXT ou NEXT I,J,K
1220 REM
1230 REM
1240 REM
1250 REM
1260 REM
1270 REM
1280 REM
1290 REM
1300 REM
1310 REM
1320 REM
1330 REM
1340 REM
1350 REM
1360 REM
1370 REM
1380 REM
1390 REM
1400 REM
1410 REM  Sur TRS 80 il n'y a pas de séparateur: les noms de variables composant FOR
1420 REM  et NEXT ne sont donc pas accolées. Le programme n'en trouve sensiblement aucune
1430 REM
1440 REM
1450 REM
1460 REM
1470 REM
1480 REM
1490 REM
1500 REM
1510 REM
1520 REM
1530 REM
1540 REM
1550 REM
1560 REM
1570 REM
1580 REM
1590 REM
1600 REM
1610 REM
1620 REM
1630 REM
1640 REM
1650 REM
1660 REM
1670 REM
1680 REM
1690 REM
1700 REM
1710 REM
1720 REM
1730 REM
1740 REM
1750 REM
1760 REM
1770 REM
1780 REM
1790 REM
1800 REM
1810 REM
1820 REM
1830 REM
1840 REM
1850 REM
1860 REM
1870 REM
1880 REM
1890 REM
1900 REM
1910 REM
1920 REM
1930 REM
1940 REM
1950 REM
1960 REM
1970 REM
1980 REM
1990 REM
2000 REM
2010 REM
2020 REM
2030 REM
2040 REM
2050 REM
2060 REM
2070 REM
2080 REM
2090 REM
2100 REM
2110 REM
2120 REM
2130 REM
2140 REM
2150 REM
2160 REM
2170 REM
2180 REM
2190 REM
2200 REM
2210 REM
2220 REM
2230 REM
2240 REM
2250 REM
2260 REM
2270 REM
2280 REM
2290 REM
2300 REM
2310 REM
2320 REM
2330 REM
2340 REM
2350 REM
2360 REM
2370 REM
2380 REM
2390 REM
2400 REM
2410 REM
2420 REM
2430 REM
2440 REM
2450 REM
2460 REM
2470 REM
2480 REM
2490 REM
2500 REM
2510 REM
2520 REM
2530 REM
2540 REM
2550 REM
2560 REM
2570 REM
2580 REM
2590 REM
2600 REM
2610 REM
2620 REM
2630 REM
2640 REM
2650 REM
2660 REM
2670 REM
2680 REM
2690 REM
2700 REM
2710 REM
2720 REM
2730 REM
2740 REM
2750 REM
2760 REM
2770 REM
2780 REM
2790 REM
2800 REM
2810 REM
2820 REM
2830 REM
2840 REM
2850 REM
2860 REM
2870 REM
2880 REM
2890 REM
2900 REM
2910 REM
2920 REM
2930 REM
2940 REM
2950 REM
2960 REM
2970 REM
2980 REM
2990 REM
3000 REM
3010 REM
3020 REM
3030 REM
3040 REM
3050 REM
3060 REM
3070 REM
3080 REM
3090 REM
3100 REM
3110 REM
3120 REM
3130 REM
3140 REM
3150 REM
3160 REM
3170 REM
3180 REM
3190 REM
3200 REM
3210 REM
3220 REM
3230 REM
3240 REM
3250 REM
3260 REM
3270 REM
3280 REM
3290 REM
3300 REM
3310 REM
3320 REM
3330 REM
3340 REM
3350 REM
3360 REM
3370 REM
3380 REM
3390 REM
3400 REM
3410 REM
3420 REM
3430 REM
3440 REM
3450 REM
3460 REM
3470 REM
3480 REM
3490 REM
3500 REM
3510 REM
3520 REM
3530 REM
3540 REM
3550 REM
3560 REM
3570 REM
3580 REM
3590 REM
3600 REM
3610 REM
3620 REM
3630 REM
3640 REM
3650 REM
3660 REM
3670 REM
3680 REM
3690 REM
3700 REM
3710 REM
3720 REM
3730 REM
3740 REM
3750 REM
3760 REM
3770 REM
3780 REM
3790 REM
3800 REM
3810 REM
3820 REM
3830 REM
3840 REM
3850 REM
3860 REM
3870 REM
3880 REM
3890 REM
3900 REM
3910 REM
3920 REM
3930 REM
3940 REM
3950 REM
3960 REM
3970 REM
3980 REM
3990 REM
4000 REM
4010 REM
4020 REM
4030 REM
4040 REM
4050 REM
4060 REM
4070 REM
4080 REM
4090 REM
4100 REM
4110 REM
4120 REM
4130 REM
4140 REM
4150 REM
4160 REM
4170 REM
4180 REM
4190 REM
4200 REM
4210 REM
4220 REM
4230 REM
4240 REM
4250 REM
4260 REM
4270 REM
4280 REM
4290 REM
4300 REM
4310 REM
4320 REM
4330 REM
4340 REM
4350 REM
4360 REM
4370 REM
4380 REM
4390 REM
4400 REM
4410 REM
4420 REM
4430 REM
4440 REM
4450 REM
4460 REM
4470 REM
4480 REM
4490 REM
4500 REM
4510 REM
4520 REM
4530 REM
4540 REM
4550 REM
4560 REM
4570 REM
4580 REM
4590 REM
4600 REM
4610 REM
4620 REM
4630 REM
4640 REM
4650 REM
4660 REM
4670 REM
4680 REM
4690 REM
4700 REM
4710 REM
4720 REM
4730 REM
4740 REM
4750 REM
4760 REM
4770 REM
4780 REM
4790 REM
4800 REM
4810 REM
4820 REM
4830 REM
4840 REM
4850 REM
4860 REM
4870 REM
4880 REM
4890 REM
4900 REM
4910 REM
4920 REM
4930 REM
4940 REM
4950 REM
4960 REM
4970 REM
4980 REM
4990 REM
5000 REM
5010 REM
5020 REM
5030 REM
5040 REM
5050 REM
5060 REM
5070 REM
5080 REM
5090 REM
5100 REM
5110 REM
5120 REM
5130 REM
5140 REM
5150 REM
5160 REM
5170 REM
5180 REM
5190 REM
5200 REM
5210 REM
5220 REM
5230 REM
5240 REM
5250 REM
5260 REM
5270 REM
5280 REM
5290 REM
5300 REM
5310 REM
5320 REM
5330 REM
5340 REM
5350 REM
5360 REM
5370 REM
5380 REM
5390 REM
5400 REM
5410 REM
5420 REM
5430 REM
5440 REM
5450 REM
5460 REM
5470 REM
5480 REM
5490 REM
5500 REM
5510 REM
5520 REM
5530 REM
5540 REM
5550 REM
5560 REM
5570 REM
5580 REM
5590 REM
5600 REM
5610 REM
5620 REM
5630 REM
5640 REM
5650 REM
5660 REM
5670 REM
5680 REM
5690 REM
5700 REM
5710 REM
5720 REM
5730 REM
5740 REM
5750 REM
5760 REM
5770 REM
5780 REM
5790 REM
5800 REM
5810 REM
5820 REM
5830 REM
5840 REM
5850 REM
5860 REM
5870 REM
5880 REM
5890 REM
5900 REM
5910 REM
5920 REM
5930 REM
5940 REM
5950 REM
5960 REM
5970 REM
5980 REM
5990 REM
6000 REM
6010 REM
6020 REM
6030 REM
6040 REM
6050 REM
6060 REM
6070 REM
6080 REM
6090 REM
6100 REM
6110 REM
6120 REM
6130 REM
6140 REM
6150 REM
6160 REM
6170 REM
6180 REM
6190 REM
6200 REM
6210 REM
6220 REM
6230 REM
6240 REM
6250 REM
6260 REM
6270 REM
6280 REM
6290 REM
6300 REM
6310 REM
6320 REM
6330 REM
6340 REM
6350 REM
6360 REM
6370 REM
6380 REM
6390 REM
6400 REM
6410 REM
6420 REM
6430 REM
6440 REM
6450 REM
6460 REM
6470 REM
6480 REM
6490 REM
6500 REM
6510 REM
6520 REM
6530 REM
6540 REM
6550 REM
6560 REM
6570 REM
6580 REM
6590 REM
6600 REM
6610 REM
6620 REM
6630 REM
6640 REM
6650 REM
6660 REM
6670 REM
6680 REM
6690 REM
6700 REM
6710 REM
6720 REM
6730 REM
6740 REM
6750 REM
6760 REM
6770 REM
6780 REM
6790 REM
6800 REM
6810 REM
6820 REM
6830 REM
6840 REM
6850 REM
6860 REM
6870 REM
6880 REM
6890 REM
6900 REM
6910 REM
6920 REM
6930 REM
6940 REM
6950 REM
6960 REM
6970 REM
6980 REM
6990 REM
7000 REM
7010 REM
7020 REM
7030 REM
7040 REM
7050 REM
7060 REM
7070 REM
7080 REM
7090 REM
7100 REM
7110 REM
7120 REM
7130 REM
7140 REM
7150 REM
7160 REM
7170 REM
7180 REM
7190 REM
7200 REM
7210 REM
7220 REM
7230 REM
7240 REM
7250 REM
7260 REM
7270 REM
7280 REM
7290 REM
7300 REM
7310 REM
7320 REM
7330 REM
7340 REM
7350 REM
7360 REM
7370 REM
7380 REM
7390 REM
7400 REM
7410 REM
7420 REM
7430 REM
7440 REM
7450 REM
7460 REM
7470 REM
7480 REM
7490 REM
7500 REM
7510 REM
7520 REM
7530 REM
7540 REM
7550 REM
7560 REM
7570 REM
7580 REM
7590 REM
7600 REM
7610 REM
7620 REM
7630 REM
7640 REM
7650 REM
7660 REM
7670 REM
7680 REM
7690 REM
7700 REM
7710 REM
7720 REM
7730 REM
7740 REM
7750 REM
7760 REM
7770 REM
7780 REM
7790 REM
7800 REM
7810 REM
7820 REM
7830 REM
7840 REM
7850 REM
7860 REM
7870 REM
7880 REM
7890 REM
7900 REM
7910 REM
7920 REM
7930 REM
7940 REM
7950 REM
7960 REM
7970 REM
7980 REM
7990 REM
8000 REM
8010 REM
8020 REM
8030 REM
8040 REM
8050 REM
8060 REM
8070 REM
8080 REM
8090 REM
8100 REM
8110 REM
8120 REM
8130 REM
8140 REM
8150 REM
8160 REM
8170 REM
8180 REM
8190 REM
8200 REM
8210 REM
8220 REM
8230 REM
8240 REM
8250 REM
8260 REM
8270 REM
8280 REM
8290 REM
8300 REM
8310 REM
8320 REM
8330 REM
8340 REM
8350 REM
8360 REM
8370 REM
8380 REM
8390 REM
8400 REM
8410 REM
8420 REM
8430 REM
8440 REM
8450 REM
8460 REM
8470 REM
8480 REM
8490 REM
8500 REM
8510 REM
8520 REM
8530 REM
8540 REM
8550 REM
8560 REM
8570 REM
8580 REM
8590 REM
8600 REM
8610 REM
8620 REM
8630 REM
8640 REM
8650 REM
8660 REM
8670 REM
8680 REM
8690 REM
8700 REM
8710 REM
8720 REM
8730 REM
8740 REM
8750 REM
8760 REM
8770 REM
8780 REM
8790 REM
8800 REM
8810 REM
8820 REM
8830 REM
8840 REM
8850 REM
8860 REM
8870 REM
8880 REM
8890 REM
8900 REM
8910 REM
8920 REM
8930 REM
8940 REM
8950 REM
8960 REM
8970 REM
8980 REM
8990 REM
9000 REM
9010 REM
9020 REM
9030 REM
9040 REM
9050 REM
9060 REM
9070 REM
9080 REM
9090 REM
9100 REM
9110 REM
9120 REM
9130 REM
9140 REM
9150 REM
9160 REM
9170 REM
9180 REM
9190 REM
9200 REM
9210 REM
9220 REM
9230 REM
9240 REM
9250 REM
9260 REM
9270 REM
9280 REM
9290 REM
9300 REM
9310 REM
9320 REM
9330 REM
9340 REM
9350 REM
9360 REM
9370 REM
9380 REM
9390 REM
9400 REM
9410 REM
9420 REM
9430 REM
9440 REM
9450 REM
9460 REM
9470 REM
9480 REM
9490 REM
9500 REM
9510 REM
9520 REM
9530 REM
9540 REM
9550 REM
9560 REM
9570 REM
9580 REM
9590 REM
9600 REM
9610 REM
9620 REM
9630 REM
9640 REM
9650 REM
9660 REM
9670 REM
9680 REM
9690 REM
9700 REM
9710 REM
9720 REM
9730 REM
9740 REM
9750 REM
9760 REM
9770 REM
9780 REM
9790 REM
9800 REM
9810 REM
9820 REM
9830 REM
9840 REM
9850 REM
9860 REM
9870 REM
9880 REM
9890 REM
9900 REM
9910 REM
9920 REM
9930 REM
9940 REM
9950 REM
9960 REM
9970 REM
9980 REM
9990 REM
```

Liste du programme (suite)

RECHERCHE SEQUENTIELLE DANS UNE TABLE

Nous rappelons tout d'abord, brièvement, qu'une « table » est un ensemble de *mots* (ici, nous avons choisi le vocabulaire d'un BASIC), ou un ensemble de *codes* (par exemple, les codes des départements français). Le problème consiste à fournir un mot, ou un code, que nous appellerons « Argument de Recherche », et à déterminer

- si l'argument fourni fait partie de la *table*,
- si c'est le cas, d'indiquer quel est le numéro d'ordre *(indice)* de ce mot dans la table

La table peut se présenter sous deux formes

- Les mots ou les codes apparaissent dans un *ordre* croissant ou décroissant (ordre alphabétique ou numérique, qui sont les deux cas les plus fréquents)
- Les mots ou les codes apparaissent dans le désordre

Il existe deux méthodes principales pour résoudre le problème

- On compare successivement l'argument avec tous les termes, du premier au dernier, jusqu'à ce que l'on ait trouvé l'égalité, ou épuisé la table
- On découpe la table successivement en deux morceaux

La première méthode, la moins rapide, s'appelle « recherche séquentielle ». Elle peut s'appliquer à n'importe quelle table, ordonnée ou non. La seconde méthode, beaucoup plus rapide et efficace, s'appelle « recherche dichotomique ». Elle ne peut s'appliquer qu'à une table ordonnée. Cette fiche présente la première méthode.

Le programme donné ici se compose de trois sous-programmes et d'un programme principal. Le sous-programme de recherche séquentielle est celui qui commence en 300. Au moment où on l'appelle, il faut que la table T$ soit dimensionnée à N9 (donc contienne N 9 + 1 valeurs), et que la variable A$ contienne la chaîne que l'on veut trouver. Le sous-programme commençant en 100 se charge de l'initialisation (de N 9 et de T$), et A$ reçoit une valeur dans le programme principal, qui sert justement à tester l'ensemble. Le sous-programme 200-299 est utilisé pour imprimer « proprement » la table TS, permettant ainsi de vérifier que l'initialisation s'est effectuée correctement.

En sortie du sous-programme de recherche, on obtient K 3 qui est le numéro de A$ dans la table. Si A$ n'est pas dans la table, K 3 vaut — 1.

Notez l'astuce utilisée pour l'impression aux lignes 230 et 240, ainsi que l'utilisation de K 3 aux lignes 320, 340 et 360. L'utilisation du test de K 3 = — 1 en 340 permet d'éviter de sortir de la boucle 330-350 par un GOTO, ce qui pourrait poser des problèmes avec les BASIC de Microsoft.

```
10 REM RECHERCHE SEQUENTIELLE DANS UNE TABLE
20 REM COPYRIGHT L'ORDINATEUR INDIVIDUEL
30 GOSUB100 REM INITIALISATION
40 REM BOUCLE DE LECTURE
50 PRINT INPUT"DONNEZ UN MOT" A$
60 IFA$="LISTE"THENGOSUB200 GOTO40
70 IFA$<>"FIN"THENGOSUB300 GOTO40
80 END
100 REM INITIALISATION
110 READN9 REM NOMBRE DE MOTS DANS LA LISTE (MOINS UN)
120 DATA49
130 DINT$(N9)
140 FORJ=0TON9 READT$(J) NEXTJ
145 REM LES N9+1 MOTS DE LA LISTE
150 DATA CLR BATA DIM END FOR GOSUB
155 DATA GOTO INPUT INT LET LIST NEW
160 DATA NEXT PRINT READ REM RETURN
165 DATA RUN SGN STN SQR STEP STOP TAB
170 DATA THEN AND CHR$ LOAD COS SAVE
175 DATA EXP LEFT$ LEN LOG MID$ OR NOT
180 DATA PEEK POKE RIGHT$ SPC STR$ TAN
185 DATA ABS INT TO RND VAL WAIT
190 RETURN
200 REM AFFICHAGE DES MOTS DE LA LISTE
210 REM LA LISTE EST DANS T$(N9)
220 FORJ=0TON9
230 A$=T$(J)+"                    " REM ON AJOUTE 18 BLANCS DEPUIS LE MOT
240 PRINTLEFT$(A$,18) REM AVANT DE N EN PRENDRE QUE LES 18 PRINTER 18 ARACTERES
250 NEXTJ
260 PRINT
299 RETURN
300 REM RECHERCHE SEQUENTIELLE DANS UNE TABLE
310 FOR I=1 TO N9
320 IF A$=T$(I)THENPRINT"CE MOT EXISTE PAS" GOTO399
330 PRINT CE MOT EST LE MOT NO." K3 "DE LA TYPE LISTE"
399 RETURN
```

GESTION DE FICHIER DIFFERENTIEL

Pour se prémunir contre des destructions de fichiers en cours de mise à jour, une solution simple (et vivement conseillée !) consiste à prévoir un fichier différentiel. Dans ce fichier, on duplique tous les enregistrements du fichier principal qui ont été modifiés. Il est ainsi possible de regénérer un fichier détruit en cours de mise à jour grâce à la dernière sauvegarde et au fichier différentiel.

La sauvegarde des enregistrements modifiés est effectuée dans un fichier en accès direct dans le programme qui suit. On pourrait également la réaliser dans un fichier séquentiel ce qui donnerait un programme légèrement différent.

Il peut être utile de prévoir une sécurité qui indique, à chaque fois que le fichier différentiel devient trop important, le moment de réaliser une sauvegarde du fichier principal et de remettre à zéro le fichier différentiel

Le programme tourne sur TRS-80. Sur d'autres systèmes, il faudra remplacer LOF(2) + 1 par LOF(2) – lignes 240-50.300.)

```
10 '                    GESTION DE FICHIER DIFFERENTIEL
20 '                       AUTEUR : JACQUES BOISGONTIER
30 '   COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
40 OPEN "R",#1,"FPRIN"
50 OPEN "R",#2,"FDIFF"
60 FIELD #1,15 AS N$,20 AS RIEN$
70 FIELD #1,126 AS I1$
80 FIELD #2,126 AS I2$,2 AS NE$
90 '--
100 INPUT "MODE? ";M$
110 IF M$="C" THEN GOSUB 160      ' CREATION
120 IF M$="REGEN" THEN GOSUB 290  ' REGENERATION DU FP
130 GOTO 100
140 '-------------------------- CREATION / MODIFICATION
150 '
160 INPUT "QUEL ENREG?";RANG
170 INPUT "NOM? ";X$:IF X$="" THEN RETURN
180 LSET I1$=STRING$(CHR$(0),126)
190 LSET N$=X$
200 PUT #1,RANG
210 '-------- SAUVEGARDE DANS FICHIER DIFFERENTIEL
220 LSET I2$=I1$
230 LSET NE$=MKI$(RANG)
240 PUT #2,LOF(2) + 1   'POUR TRS 80
250 PRINT "LOF(2) + 1 =";LOF(2) + 1     '
260 GOTO 170
270 '-------- REGENERATION DU FICHIER PRINCIPAL
280 '
290 PRINT:INPUT "AVEZ-VOUS REMPLACE LE FICHIER PRINCIPAL
    PAR LA DERNIERE SAUVEGARDE? ";R$
300 FOR I=1 TO LOF(2) + 1
310   GET #2,I:IF CVI(NE$)=0 THEN 360
320   X=CVI(NE$)
330   LSET I1$=I2$
340   PUT #1,X
350   PRINT "X=";X;N$
360 NEXT I
370 RETURN
380 '-------- LA SAUVEGARDE DES ENREGISTREMENTS MODIFIES
390 'POURRAIT ETRE FAITE DANS UN FICHIER SEQUENTIEL
```


la fin du déclin:

l'ordinateur retrouve votre latin

Qui ne se souvient de la chanson de Jacques Brel « Rosa Rosa Rosam... Rosarum Rosis Rosis » ? Cet énoncé faisait alors apparaître le latin comme une discipline bien plus avenante que ne le laissaient supposer certains énormes dictionnaires... Car si le latin est dit « langue morte », il reste cependant bien vivant en de nombreux endroits ainsi qu'en témoigne cette rencontre avec l'informatique individuelle.

Après avoir beaucoup souffert avec l'apprentissage des cinq déclinaisons que comporte le latin, on se rend compte qu'elles obéissent finalement à des règles très strictes. Il ne reste alors qu'à franchir ensuite le pas qui consiste à se dire « mais alors pourquoi ne pas l'automatiser » On peut obtenir ainsi un programme qui à partir du nominatif et du génitif d'un mot en donne la déclinaison

Ces deux renseignements sont les seuls dont le programme ait a priori besoin pour fournir l'ensemble de la déclinaison ainsi que son numéro (première, troisième) Neanmoins, il peut y avoir doute Dans le cas de la troisième déclinaison par exemple, le programme demande alors le genre du mot, puis, à partir de ce renseignement, détermine s'il s'agit d'un impari (ou pari) syllabique avant de fournir la déclinaison

L'intérêt de ce programme est donc qu'il traite effectivement tous les cas avec une information de départ minimum On trouvera ci-contre la liste ainsi qu'un exemple d'exécution

Qui a dit que le latin était mort ? Fluctuat nec mergitur !

Alain Richard

en classe

comptez-bien sur votre ordinateur

En classe de cinquième, le programme de mathématiques prévoit l'étude des nombres entiera. L'enseignant est amené è introduire de nombreuses notions nouvelles : diviseurs et multiples d'un nombre, diviseurs communs et multiples communs de deux nombres, et finalement nombres premiers et factorisation première d'un nombre. Cette dernière est une notion-clé lors de la recherche de tous les diviseurs d'un nombre ou lors du calcul du plus grand diviseur commun (PGDC) ou du plus petit commun multiple non nul (PPCM). Tous ces problèmes sont résolus par des règles de calcul assez simples. Pour l'élève, la seule façon de les assimiler est de faire de nombreux exercices. L'enseignant, lui, n'a pas à refaire tous ces calculs ; seul le résultat l'intéresse. Quoi de plus simple alors que d'écrire le petit programme en BASIC présenté ici ?

Avent d'en erriver au programme proprement dit, rappelons quelques notions fondamantales

Un nombre premier est un nombra qui admet comme seuls diviseurs 1 et lui-même

Per exemple, les diviseurs de 7 sont 1 et 7 donc 7 est premier ; les diviseurs de 9 sont 1, 3 et 9 donc 9 n'est pes premier

La factorisation première d'un nombre est la décomposition de ce nombre en un produit de nombres (ou facteurs) premiers, soit par exemple 100 = 2*2*5*5, 40 = 2*2*2*5

Rappelons à l'aide d'un exemple comment on trouve le PGCD et le PPCM de deux nombras

D'abord il faut décomposer ces nombres en facteurs premiers

Le PGCD est le produit des facteurs communs aux deux nombres
PGCD (40,100) = 2*2*5 = 20

Le PPCM ast le produit des facteurs epparaissant au moins une fois dans les deux nombres PPCM (40,100) = 2*2*2*5*5 = 200

Il est utile de remarquer que 40*100 = PGCD (40,100) * PPCM (40, 100)

La méthode ci-dessus est celle qui est enseignée, car c'est la plus simpla, mais ce n'est pas la plus rapide Considérons l'exemple suivant
100 = 40*2 + 20
40 = 20 * 2 + 0

On divise le grend nombre (100) par le petit (40) Tant que le reste de la division n'est pas nul, on divise le diviseur par le reste Lorsque le reste est nul, alors le PGCD est le dernier diviseur obtenu

Autre exemple PGCD(26,84) =
84 = 26* 3+6
26 = 6* 4+2
6 = 2* 3+0

PGCD (26,84) = 2

Ayant le PGCD on a immédiatement la PPCM
PPCM (a,b) = a*b/PGCD (a,b)

Maintenant nous sommes prêts à eborder le programme proposé

La partie centrale du programme qui comprend les lignes 100-250, est une boucle infinie En fonction du problème choisi, l'ordinateur continue avec le sous-programme correspondant (ligne 220) Ensuite, il recommence dès le début ou avec le même sous-programme (230-250)

Les sous-programmes 1 000-2 250 indiqués à la ligne 220 sont en fait des routines abordant le problème Les calculs sont effectués dans les sous-programmes commençant aux lignes 2 600, 2 800, 2 900, 3 100 et 3 200

Remerquons que ces sous-programmes sont les plus importants et cependant ils représentent moins d'un quart du programme D'autre part il ne faut pas sous-estimer le rôle des calculs préliminaires dans les routines 1 000-2 250 Ils permettent de rendre les routines de calcul beaucoup plus efficaces

Par exemple, à la ligne 1 120 on vérifie que le nombre est impair, sinon il est inutile de rechercher dans la routine 2 600 s'il est premier

L'adaptation à d'autres ordinateurs ne présente que peu de difficultés Le programme utilise souvent la fonction INKEY $ Celle-ci est uniquement utilisée par commodité et peut donc être remplacée par INPUT

Finalement PRINT @ 128, écrit en haut de l'écran et PRINT @ 832, écrit en bas de l'écran

Si vous cherchez les diviseurs d'un nombre qui en a plus de 100, il vous faudre modifier la ligne 100 en fonction de vos besoins

Jean-Paul Zanter

liste de programme sur Pet

exemple d'exécution

Premiere parution L OI n° 27

Liste du programme

```
10 REM          * LES MATHEMATIQUES A L'ECOLE *
20 REM  AUTEUR JEAN-PAUL ZANTAN
30 REM  COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
40 REM*****************************************************************
100 DIM D(50),Q(50)
110 CLS
120 PRINT@318,
130 PRINT TAB(16),"1 - NOMBRES PREMIERS"
140 PRINT TAB(14),"2 - PGCD ET PPCM"
150 PRINT TAB(14),"3 - DIVISEURS"
160 PRINT TAB(14),"4 - DIVISEURS COMMUNS"
170 PRINT TAB(14),"5 - MULTIPLES"
180 PRINT TAB(14),"6 - MULTIPLES COMMUNS"
190 PRINT TAB(14),"7 - FACTORISATION PREMIERE"
200 GOSUB 2300          : REM CHOIX D'UN PROBLEME
210 CLS
220 ON PC GOSUB 1300,1200,1100,1500,1700,1900,2100
230 PRINT @832,"AVEZ-VOUS UN AUTRE PROBLEME SEMBLABLE?"
240 GOSUB 9000          : REM TEST SUR LA REPONSE
250 IF RS="O" THEN 210 ELSE 110
1000 REM*****************************************************************
1010 PRINT "LISTE DES NOMBRES PREMIERS DE",A,"A",
1020 GOSUB 2400         : REM LECTURE DES DEUX NOMBRES
1030 CLS
1040 PRINT"LISTE DES NOMBRES PREMIERS ENTRE",MIN,"ET",MAX
1050 PRINT
1060 IF MIN > 3 THEN 1120
1070 IF MAX > 3 THEN Z=3 ELSE Z=MAX
1080 FOR X=MIN TO Z
1090     PRINT X,
1100 NEXT X
1110 MIN=3
1120 IF MIN=INT(MIN/2)*2 THEN IF MIN=MAX THEN 1140 ELSE MIN=MIN+1
1130 GOSUB 2600
1140 RETURN
1200 REM*****************************************************************
1210 PRINT "CALCUL DE PGCD ET DU PPCM DE DEUX NOMBRES"
1220 GOSUB 2400         : REM LECTURE DES DEUX NOMBRES
1230 A=MAX: B=MIN
1240 GOSUB 2800         : REM CALCUL DU PGCD ET DU PPCM
1250 CLS
1260 PRINT@334,"PGCD(",MIN,",",MAX,")=",PGCD
1270 PRINT@452,"PPCM(",MIN,",",MAX,")=",PPCM
1280 RETURN
1300 REM*****************************************************************
1310 PRINT "RECHERCHE DES DIVISEURS D'UN NOMBRE"
1320 GOSUB 2500         : REM LECTURE DU NOMBRE
1330 CLS
1340 PRINT"LISTE DES DIVISEURS DE",MIN,"PRINT
1350 GOSUB 1400
1360 RETURN
1400 REM*****************************************************************
1410 IF MIN=1 THEN PRINT MIN : GOTO 1440
1420 IF MIN=2 THEN PRINT "1",MIN : GOTO 1440
1430 A=MIN: GOSUB 2900: REM CALCUL DES DIVISEURS
1440 RETURN
1500 REM*****************************************************************
1510 PRINT "RECHERCHE DES DIVISEURS COMMUNS DE DEUX NOMBRES"
1520 GOSUB 2400         : REM LECTURE DES DEUX NOMBRES
1530 CLS
1540 PRINT"LISTE DES DIVISEURS COMMUNS DE",MIN,"ET",MAX
1550 PRINT
1560 A=MAX: B=MIN
1570 GOSUB 2800         : REM CALCUL DES DIVISEURS COMMUNS
1580 MIN=PGCD
1590 GOSUB 1400         : REM RECHERCHE DES DIVISEURS
1600 RETURN
1700 REM*****************************************************************
1710 PRINT "RECHERCHE DES MULTIPLES D'UN NOMBRE"
1720 GOSUB 2500         : REM LECTURE DU NOMBRE
1730 A=MIN
1740 PRINT @768,"INPUT"COMBIEN DE MULTIPLES",NLM
1750 B=MIN: GOSUB 2500
1760 CLS
1770 PRINT"LISTE DES",B,"PREMIERS MULTIPLES DE",A
1780 PRINT
1790 GOSUB 3100         : REM CALCUL DES MULTIPLES
1800 RETURN
1900 REM*****************************************************************
1910 PRINT "RECHERCHE DES MULTIPLES COMMUNS DE DEUX NOMBRES"
1920 GOSUB 2400         : REM LECTURE DES DEUX NOMBRES
1930 A=MAX: B=MIN
1940 PRINT @768,"INPUT"COMBIEN DE MULTIPLES",NLM
1950 GOSUB 2530
1960 CLS
1970 PRINT"LISTE DES",NLM,"PREMIERS MULTIPLES COMMUNS DE",B,"ET DE",A
1980 GOSUB 2800         : REM CALCUL DU PPCM
1990 A=PPCM
2000 B=MIN
2010 GOSUB 3100         : REM CALCUL DES MULTIPLES
2020 RETURN
2100 REM*****************************************************************
2110 PRINT "FACTORISATION PREMIER D'UN NOMBRE"
2120 GOSUB 2500         : REM LECTURE DU NOMBRE
2130 NF=MIN
2140 IF NF=1 THEN CLS
                 PRINT @512,"1 NE PEUT ETRE FACTORISE"
                 : GOTO 2270
2150 CLS
2160 PRINT"FACTORISATION PREMIERE DE",NF
2170 PRINT
2180 A=2 : GOSUB 3200
2190 IF NF=1 THEN 2270
2200 TT=0
2210 FOR A=3 TO SQR(NF) STEP 2
2220     GOSUB 3200
2230     IF TT=1 THEN A=SQR(NF) : TT=1
2240 NEXT A
2250 IF TT=1 THEN 2270
2260 PRINT NF,"EXPOSANT 1"
2270 RETURN
2300 REM*****************************************************************
2310 REM          CHOIX D'UN PROBLEME
2320 PRINT @768,"QUEL PROBLEME VOULEZ-VOUS RESOUDRE?"
2330 PC$=INKEY$
2340 IF PC$=" " THEN 2330
2350 PC=VAL(PC$)
2360 IF PC=0 THEN 2320
2370 RETURN
2400 REM*****************************************************************
2410 REM          LECTURE DES DEUX NOMBRES
2420 PRINT@768,"INPUT"INTRODUIRE LES DEUX NOMBRES ",MIN,MAX
2430 IF MIN<>INT(MIN) OR MAX<>INT(MAX) THEN CLS
                                 : PRINT @768,"LES NOMBRES DOIVENT ETRE ENTIERS"
                                 : GOTO 2420
2440 IF MIN>MAX THEN Q=MIN : MIN=MAX : MAX=Q
2450 IF MIN<=0 THEN CLS
                 : PRINT@768,"LES NOMBRES DOIVENT ETRE STRICTEMENT POSITIFS"
                 : GOTO 2420
2460 RETURN
2500 REM*****************************************************************
2510 REM          LECTURE D'UN NOMBRE
2520 PRINT@768,"INPUT"INTRODUIRE LE NOMBRE",MIN
2530 IF MIN<>INT(MIN) THEN CLS
                                 : PRINT@768,"LE NOMBRE DOIT ETRE ENTIER"
                                 : GOTO 2520
2540 IF MIN<=0 THEN CLS
                 : PRINT@768,"LE NOMBRE DOIT ETRE STRICTEMENT POSITIF"
                 : GOTO 2520
2550 RETURN
2600 REM*****************************************************************
2610 REM          RECHERCHE DES NOMBRES PREMIERS ENTRE MIN ET MAX
2620 FOR A=MIN TO MAX STEP 2
2630     ZZ=0
2640     FOR B=3 TO SQR(A) STEP 2
2650         IF A=INT(A/B)*B THEN ZZ=ZZ+1 : B=SQR(A)
2660     NEXT B
2670     IF ZZ=1 THEN 2690
2680     PRINT A,
2690 NEXT A
2700 RETURN
2800 REM*****************************************************************
2810 REM          RECHERCHE DU PGCD ET DU PPCM DE A ET DE B (A>B)
2820 PRODUIT=A*B
2830 IF B=0 THEN PGCD=A: PPCM=PRODUIT/A : GOTO 2870
2840 R=A-INT(A/B)*B
2850 A=B : B=R
2860 GOTO 2830
2870 RETURN
2900 REM*****************************************************************
2910 REM          RECHERCHE DES DIVISEURS DE A
2920 N=1 : D(1)=1 : Q(1)=A
2930 FOR B=2 TO SQR(A)
2940     R=A-INT(A/B)*B
2950     IF R=0 THEN N=N+1 : D(N)=B : Q(N)=A/B
2960 NEXT B
2970 FOR Z=1 TO N : PRINT D(Z), : NEXT Z
2980 IF D(N)=Q(N) THEN N=N-1
2990 FOR Z=N TO 1 STEP-1 : PRINT Q(Z), : NEXT Z
3000 RETURN
3100 REM*****************************************************************
3110 REM          CALCUL DES N PREMIERS MULTIPLES DE A
3120 FOR N=0 TO N-1 : PRINT A*N, : NEXT N
3130 RETURN
3200 REM*****************************************************************
3210 REM          CALCUL DES FACTEURS PREMIERS DE NF
3220 ND=0
3230 Q=NF/A
3240 IF Q=INT(Q) THEN ND=ND+1 : NF=Q : GOTO 3230
3250 IF ND>0 THEN PRINT A,"EXPOSANT",ND
3260 RETURN
9000 REM*****************************************************************
9010 REM          TEST SUR LA REPONSE
9020 RS=INKEY$
9030 IF RS="" THEN 9020
9040 IF RS<>"O" AND RS<>"N" THEN PRINT"OUI OU NON" : GOTO 9020
9050 RETURN
```


antiquité binaire :
la multiplication égyptienne

Imaginez une société ne connaissant que la table de multiplication par deux. Peut-elle dans ces conditions effectuer le produit de deux nombres quelconques ? L'histoire de l'Egypte sncienne nous montre brillemment que ce problème peut être résolu ssns difficultés.

```
MULTIPLICANDE =  234                MULTIPLICATEUR =  345

          234                  ......................... 345
          117                  690
           58                  ......................... 1380
           29                  2760
           14                  ......................... 5520
            7                  11040
            3                  22080
            1                  44160

                               -------------
                               80730
```

Nous désirons donc multiplier le nombre X par le nombre Y (X et Y étant entiers).

En partant de X (de rang n égal à 1), il faut tout d'abord dresser une première colonne de nombres, tels que chacun d'eux soit égal à la partie entière du rapport $X_{n-1}/2$. Pour obtenir le 3e nombre X_3 on partira donc du deuxième X_2 et on aura $X_3 = INT(X_2/2)$. Restant dans des multiplications de nombres entiers positifs il suffit de s'arrêter quand X_n veut 1.

Puis en partant de Y (de rang n égal à 1), on dresse une deuxième colonne de nombres, tels que chacun d'eux soit égal au double de celui qui la précède. Pour des raisons de présentation, on affichera ou on imprimera sur le même ligne les nombres des deux colonnes ayant même rang.

On peut maintenant effacer (supprimer si on en tire une liste) les nombres de la deuxième colonne pour lesquels les nombres de même rang de la première colonne sont pairs.

Il ne reste plus qu'à effectuer la somme des nombres restants de la deuxième colonne et alors, miracle, cette somme est égale au produit racherché. Aucune alchimie n'a présidé à l'obtention de ce résultat pourquoi ne reprendriez vous pas vous-même la démonstration mathématique ?

Adapter ce programme sur votre ordinateur préféré ne posera aucun problème, les ordres PRINT@ correspondent uniquement à une présentation agréable sur l'écran. Quant à l'adapter pour déchiffrer les hiéroglyphes, c'est bien sûr une autre histoire.

Didier Heroux

Première parution : L'OI n° 27

à vos postes pour la multiplication

n'ayez plus peur des facteurs

Le jeu constitue la meilleur façon d'apprendre. Plutôt que de réciter en ânonnant 1 × 1 = 1, 1 × 2 = 2... pourquoi ne pas rendre attrayant l'apprentissage de tables de multiplication ? Ce petit programme s'y efforce : entraînez-vous au calcul mental.

Attention ! Il n'est simple et facila que dans la forme présente où les multiplications ne vont que jusqu'á 9 × 9. Il suffit de changer cinq lignes pour traiter des multiplications jusqu'á 20 × 20 ou 30 × 30 . mais avant ces développements, quel est le principe du programme ?

Il nous invite á jouer avec lui pour donner la bonne réponse á des multiplications qui deviennent de plus en plus compliquées au fur et á mesure que vous progressez. De plus en plus compliquées si vous avez répondu correctement, car si vous vous trompez, la sanction est simple : une faute et vous restez au même niveau, deux fautes ou plus et vous connaissez la déchéance de devoir repartir á un niveau inférieur. Pour arriver á la fin du programme, il vous faudra donc arriver á donner trois bonnes réponses au niveau le plus élevé.

Jusque-lá rien de bian compliqué. mais on peut corser un peu le jeu en considérant que vous ne disposaz que d'un certain temps imparti pour donner la bonne réponse. Si vous dépassez ce temps, le programme considère impitoyablement que vous avez fait une faute : tout cala est bien arbitraire, me direz-vous. Oui, bien sûr, mais vous aurez quand même le choix de votre temps de réponse entre 5 et 15 secondes. Et si 1 × 1 en 15 secondes est simple, il n'en est pas de même pour 17 × 18 en 5 secondes, par exemple.

Voilá donc les régles du jeu que vous devrez suivre. Pour parvenir á ce résultat, le programme est construit selon les principes de la programmation structurée : le programme principal (plage de numéros de lignes de 100 á 990) appelle successivement un sous-programme de début (lignes 1 000 á 1 990), un sous-programme principal (lignes 2 000 á 2 990) et un sous-programme de fin (lignes 3 000 á 3 990). La variable A est l'indicateur de fin de traitement : tant qu'elle a pour valeur zéro, on exécute le traitement principal ; quand elle prend la valeur 1, on va exécuter le traitement de fin et on arrête le programme. Catte variable ne prendra la valeur 1 que lorsque le programme aura vu que vous avez donné trois bonnes réponses. Le nombra da bonnes réponses est mémorisé dans la variable P ou 8 au niveau le plus élevé. Cette dernière variable varie donc de 1, niveau le plus faible, á 3, niveau le plus élevé, pour ne passer á 4 que lorsqu'il y a au trois bonnes réponses au niveau 3.

Le niveau de difficulté d'une multiplication est mémorisé dans le tableau D, tableau a deux dimensions (multiplicande et multiplicateur) : les valeurs du tableau sont donc 1, 2 ou 3.

Chaque question ast posée avec un multiplicanda – variable Y – et un multiplicateur – variabla Z – choisis au hasard (variables X et C). La réponse que vous donnerez sera stockée dans la variable W.

On mémorise dans le tableau E les résultats des multiplications précédemment posées, da façon á ne pas avoir trois multiplications ayant le même résultat lors d'une même série de questions.

Le temps alloué pour la réponse est stocké dans la variable T ! et contrôlé, á l'aida des variablas U et V, valeurs de l'horloge interne avant et aprés la réponse. La variable Q indique par la valeur 1 que le temps a été dépassé.

Si maintenant vous désirez rendre ce programme plus compliqué, posant par exemple des questions jusqu'á 20 × 20 ou 30 × 30 il ne vous reste qu'á modifier :

. les lignes de DATA – aprés 10 000 . mettre pour chaque multiplication, l'indication du niveau de difficultés de 1 á 3,

. les lignes 1270 et 1290 : ramplacer 9 par 20 ou 30,

. les lignes 2110 et 2140 : ramplacer 10 par 20 ou 30.

Une autre façon d'utiliser ce programme consiste par exemple á s en servir comme d'un répétiteur d'anglais ou d'allemand en mémorisant en DATA ou dans un fichier les mots anglais ou allemands, et laur traduction en français puis en les choisissant de la même manière au hasard.

En attendant, bon courage avec les multiplications. Personnellement, arriver á résoudre 23 × 27 en 5 secondes est un pari que je me garde bian da prendra.

Jean-Pierre Brunerie

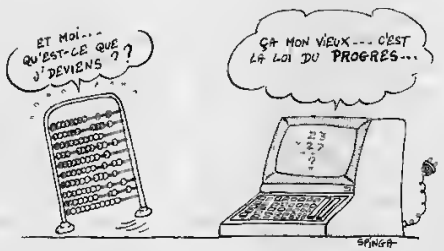
Liste du programme

```
10 REM    * MULTIPLICATION DANS L'ARCIENNE EGYPT *
20 REM AUTEUR: DIDIER GERONX
30 REM COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
40 REM*****************************************************
100 DIM X(1000), Y(1000)
110 GOSUB 1000: REM PRESENTATION
120 CLS
130 Y4=0
140 PRINT"                INTRODUCTION DES DONNES": PRINT: PRINT: PRINT
150  INPUT"  . MULTIPLICANDE = ";C: X(1)=C
160 PRINT
170 IF C<2 OR C>99999 THEN GOSUB 1200: GOTO 120
180  PRINT: PRINT
190 INPUT " . MULTIPLICATEUR = ";D1
200 IF D1<=1  THEN 190
210 Y(1)=D1
220 REM TEST POUR MULTIPLICATION TROP GRANDE
230 IF C*D1>1E+09 THEN GOSUB 1300: GOTO 150
240 CLS
250 PRINT
260 PRINTTAB(9,X(1);PRINTTAB(4,Y(1);
270 Y4=Y(1): ZZ=0: Z=0
280 FOR I=2 TO 1000
290      X(I)=INT(X(I-1)/2)
300      Y(I)=Y(I-1)*2
310      PRINTTAB(5+4*I,X(I);PRINTTAB(24+4*I,Y(I);
320      Y4=Y4+Y(I)
330      IF X(I)=1 THEN ZZ=1: Z=1: I=1000
340 NEXT I
350 IF ZZ=1 THEN I=Z
360 FOR J=1 TO I
370      IF X(J)/2=INT(X(J)/2) THEN Y4=Y4-Y(J):
                                 PRINTTAB(24+4*J,"...................";Y(J);
380 NEXT J
390 PRINTTAB(16,Y4;
400 FOR T=1 TO 2000:NEXT T
410 CLS
420 PRINT:PRINT
430 PRINT"                DESIRE VOUS FAIRE UNE AUTRE "
440 PRINT"                OPERATION ?":PRINT: PRINT
450 PRINT"                                                  O OU N"
460 GOSUB 9000
470 IF A$="O" THEN 120
480 END
1000 REM==========================================================
1010 REM                                  S/P PRESENTATION INITIALE
1020 CLS
1030 PRINT: PRINT
1040 PRINT"                                  . MULTIPLICATION"
1050 PRINT: PRINT
1060 PRINT"                                                  DANS"
1070 PRINT: PRINT
1080 PRINT"                                          L'EGYPTE ANCIENNE ."
1090 PRINT: PRINT: PRINT
1100 FOR T=1 TO 3000: NEXT T
1110 RETURN
1200 REM-----------------------------------------------------------
1210 REM                                  S/P RESPECT DES BORNES
1220 CLS
1230 PRINT: PRINT
1240 PRINT"                                  . ATTENTION ! LE MULTIPLICANDE"
1250 PRINT"                                    DOIT ETRE SUPERIEUR A 2 ET"
1260 PRINT"                                    INFERIEUR A 99999"
1270 FOR T=1 TO 2000: NEXT T
1280 RETURN
1300 REM-----------------------------------------------------------
1310 REM                                  S/P MESSAGE D'ERREUR
1320 PRINT: PRINT
1330 PRINT"                                  . ATTENTION ! LE PRODUIT NE-DOIT PAS"
1340 PRINT"                                    ETRE SUPERIEUR A ;"
1350 PRINT
1360 PRINT"                                                  999 999 999"
1370 PRINT: PRINT
1380 RETURN
9000 REM-----------------------------------------------------------
9010 A=1                                  S/P SAISIE D'UN CARACTERE EGAL A O(OUI) OU N(NON)
9020 I INPUT A$
9030 IF A$<>"O" AND A$<>"N" THEN 9020
9040 RETURN
```


Premiere parution : L'OI n° 30

Liste du programme sur PET

```
10 REM      * APPRENDRE LES MULTIPLICATIONS *
20 REM ( JEAN PIERRE BRUN )
30 REM COPYRIGHT ( ) DIAMANTS INDIVIDUEL ET L'A
40 REM*****************************************
100 GOSUB 1000
110 IF A=1 THEN 140
120 GOSUB 2000
130 GOTO 110
140 GOSUB 3000
150 END
1000 REM*****************************************
1010 REM          INITIALISATIONS
1020 PRINT BONJOUR
1030 PRINT
1040 GOSUB 4000
1050 PRINT
1060 PRINT "TES-VOUS FORT-F,MOYEN-M,NUL-N"
1070 INPUT R$
1080 IF R$="" THEN 1060
1090 IF R$="F" THEN B=3: GOTO 1130
1100 IF R$="M" THEN B=2: GOTO 1130
1110 IF R$="N" THEN B=1: GOTO 1130
1120 GOTO 1060
1130 PRINT"ETES-VOUS RAPIDE-R,MOYEN-M,LENT-L"
1140 INPUT R$
1150 IF R$="" THEN 1130
1160 IF R$="R" THEN T1=300: GOTO 1200
1170 IF R$="M" THEN T1=600: GOTO 1200
1180 IF R$="L" THEN T1=900: GOTO 1200
1190 GOTO 1130
1200 INPUT"UN NOMBRE,SVP",C
1210 REM------
1220 FOR I=1 TO C
1230 X=RND(X+C)
1240 NEXT I
1250 REM------
1260 REM------
1270 FOR I=1 TO 9
1280 REM---
1290 FOR J=1 TO 9
1300 READ D(I,J)
1310 NEXT J
1320 REM---
1330 NEXT I
1340 REM------
1350 RETURN
2000 REM-----------------------------------------
2010 REM          LA CALCULATION DE LA MULTIPLICATION
2020 REM------
2030 FOR I=1 TO 3
2040 E(I)=0
2050 NEXT I
2060 REM------
2070 F=0
2080 REM------
2090 FOR I=1 TO 3
2100 X=RND(X+C)
2110 Y=INT(X*10)
2120 IF Y=0 THEN 2100
2130 X=RND(X+C)
2140 Z=INT(X*10)
2150 IF Z=0 THEN 2130
2160 IF D(Y,Z)=0 THEN 2190
2170 PRINT PATINCL
2180 GOTO 2100
2190 E(I)=Y*Z
2200 IF I=1 THEN 2240
2210 IF I=2 THEN 2230
2220 IF E(I)=E(I-2) THEN 2100
2230 IF E(I)=E(I-1) THEN 2100
2240 PRINT Y;"MULTIPLIE PAR";Z;"LCAD
2250 Q=0
2260 U=TI
2270 INPUT G
2280 V=TI
2290 IF V-U>T1 THEN PRINT QUE C'EST LENT":Q=1
2300 IF F(I)=W THEN F=F+1:PRINT"BRAVO"GOTO 2320
2310 PRINT FAUX.";Y;" FOIS ";Z;"EGAL ";E(I)
2320 IF Q=1 AND E(I)=W THEN F=F-1
2330 NEXT I
2340 REM------
2350 IF F=3 THEN B=B+1
2360 IF B=4 THEN A=1:GOTO 2440
2370 IF B=0 OR B=1 THEN B=B-1
2380 IF B=0 THEN B=1
2390 PRINT
2400 IF B=3 THEN PRINT"VOUS ETES FORT"
2410 IF B=2 THEN PRINT"VOUS ETES MOYEN"
2420 IF B=1 THEN PRINT"VOUS ETES NUL"
2430 PRINT
2440 RETURN
3000 REM-----------------------------------------
3010 REM          FIN DU CALVAIRE
3020 PRINT
3030 PRINT"BRAVO, VOUS ETES TRES FORT"
3040 RETURN
4000 REM-----------------------------------------
4010 REM          PRESENTATION
4020 PRINT APPRENONS DONC LES MULTIPLICATIONS
4030 PRINT
4040 PRINT VOUS AVEZ SUPPOSE ETRE
4050 PRINT - FORT - LETTRE F
4060 PRINT - MOYEN- LETTRE M
4070 PRINT - NUL  - LETTRE N
4080 PRINT"CECI AU DEBUT DU JEU"
4090 PRINT
4100 PRINT PUIS VOUS ETES SUPPOSE REAL
4110 PRINT - RAPIDE- LETTRE R
4120 PRINT - MOYEN - LETTRE M
4130 PRINT - LENT  - LETTRE L
4140 PRINT
4150 PRINT"A CHAQUE FOIS, ON VOUS POSE"
4160 PRINT"TROIS MULTIPLICATIONS"
4170 PRINT
4180 PRINT TROIS BONNES REPONSES: VOUS PROGRESSEZ
4190 PRINT DEUX  VOUS RESTEZ AU MEME NIVEAU
4200 PRINT"ZERO OU UNE  VOUS REGRESSEZ"
4210 PRINT
4220 PRINT AVEZ-VOUS BIEN COMPRIS
4230 GOSUB 9000
4240 IF R$="Y" THEN 4290
4250 PRINT
4260 PRINT ALORS  REFLISEZ ATTENTIVEMENT
4270 PRINT
4280 GOTO 4000
4290 PRINT
4300 RETURN
9000 REM-----------------------------------------
9010 REM          TEST SUR LA REPONSE
9020 INPUT R$
9030 IF R$<> "Y" AND R$<>"N" THEN 9020
9040 RETURN
9050 REM-----------------------------------------
10000 DATA 1,1,1,1,1,1,1,1,1
10010 DATA 1,1,1,1,1,1,2,2,2
10020 DATA 1,1,1,2,1,2,3,2,3
10030 DATA 1,1,2,2,1,2,3,2,3
10040 DATA 1,1,1,1,1,1,2,2,2
10050 DATA 1,1,2,2,1,2,3,3,3
10060 DATA 1,2,3,3,2,3,3,3,3
10070 DATA 1,2,2,2,2,3,3,3,3
10080 DATA 1,2,3,3,2,3,3,3,3
```



testez vos connaissances en géographie

Arras. Chef-lieu ? Pas-de-Calais... Cette façon d'apprendre la géographie se pratiquait, il y a maintenant de nombreuses années, pour le certificat d'études primaires. Apprendre en jouant, et de plus sur un ordinateur (ZX-81 en l'occurrence) ne procure-t-il pas plus de plaisir ?

Pour un enfant, pouvoir agir sur une image représentée sur un écran de télévision est très stimulant. La passage d'une information se fait très simplement, en douceur, au propre rythme de celui qui interagit avec le programme présent dans l'ordinateur.

Le programme que nous vous proposons teste la connaissance de la situation de villes sur le carte da France. Cette carte se dessine sur l'écran et un texte défile expliquant le jeu Un plot clignote è l'emplacement d'una ville et deux noms numérotés s'inscrivent.

Transportez-vous sur votre ZX-81, en Ethiopie, au Kenya, en Arabie

L'ordinateur demende alors le réponse, la contrôle et donne son verdict : « correct » ou « faux », puis affiche le scora. En cas de mauvaise réponse, le nom da la ville s'inscrit.

A le fin du jeu, soit apràs vingt quastions, l'ordinateur commente le scora et invite è rejouer.

S'il peut pereître évident que Bastia ne se trouve pas en Bratagne, situer Tours par rapport au Mans peut être plus difficile. De plus, vous pouvez eisément modifier le programme et ajouter des noms de villes moins connues à la liste, pour eccroître la difficulté.

Première parution L OI n° 48

Programme de géographie (début)

```
   1 REM GEOGRAPHIE LA FRANCE
   5 REM (C)RENE LAGACHE
  10 PRINT TAB 12,CHR$ 5,CHR$ 3,
CHR$ 4
  15 REM ----------------------
  20 PRINT TAB 11,CHR$ 133,TAB 1
4,CHR$ 134,CHR$ 4
  25 REM ----------------------
  30 PRINT TAB 11,CHR$ 133,TAB 1
5,CHR$ 134,CHR$ 131
  35 REM ----------------------
  40 PRINT TAB 6,CHR$ 133,CHR$ 1
32,TAB 9,CHR$ 135,CHR$ 3;CHR$ 1,
TAB 17;CHR$ 134,CHR$ 131
  45 REM ----------------------
  50 PRINT TAB 6,CHR$ 2,CHR$ 8,C
HR$ 3,CHR$ 131,CHR$ 1,TAB 19;CHR
$ 3,CHR$ 134,CHR$ 4
  55 REM ----------------------
  60 PRINT TAB 2;CHR$ 131,CHR$ 1
31,CHR$ 5,CHR$ 131;CHR$ 131,CHR$
 5,TAB 21,CHR$ 133
  65 REM ----------------------
  70 PRINT TAB 2;CHR$ 129;TAB 21
.CHR$ 133
  75 REM ----------------------
  80 PRINT TAB 2,CHR$ 2,CHR$ 3,C
HR$ 131,TAB 21,CHR$ 5
  85 REM ----------------------
  90 PRINT TAB 5,CHR$ 134,CHR$ 4
 TAB 20,CHR$ 135,CHR$ 1
  95 REM ----------------------
 100 PRINT TAB 6,CHR$ 5,TAB 19,C
HR$ 135,CHR$ 1
 105 REM ----------------------
 110 PRINT TAB 6,CHR$ 134,CHR$ 4
.TAB 19,CHR$ 134
 115 REM ----------------------
 120 PRINT TAB 7,CHR$ 133,TAB 19
 CHR$ 2,CHR$ 134
 125 REM ----------------------
 130 PRINT TAB 7,CHR$ 133,TAB 20
.CHR$ 133
 135 REM ----------------------
 140 PRINT TAB 7,CHR$ 6,CHR$ 4,T
AB 20,CHR$ 135,CHR$ 1
 145 REM ----------------------
 150 PRINT TAB 7,CHR$ 5.CHR$ 1;T
AB 20,CHR$ 134
 155 REM ----------------------
 160 PRINT TAB 7,CHR$ 5,TAB 20.C
HR$ 133,TAB 24,CHR$ 135
 165 REM ----------------------
 170 PRINT TAB 7 CHR$ 5,TAB 21,C
HR$ 132,TAB 23,CHR$ 6,CHR$ 132
 175 REM ----------------------
 180 PRINT TAB 6,CHR$ 135,CHR$ 1
,TAB 21,CHR$ 5,TAB 23,CHR$ 5,CHA
$ 133
 185 REM ----------------------
 190 PRINT TAB 6,CHR$ 133,TAB 14
 CHR$ 135,CHR$ 5,CHR$ 3,CHR$ 3,C
HR$ 134,CHR$ 4,CHR$ 5,TAB 23,CHP
$ 134,CHR$ 6
 195 REM ----------------------
 200 PRINT TAB 5,CHR$ 2,CHR$ 3;C
HR$ 134,CHR$ 131,TAB 14,CHR$ 7.T
AB 19,CHR$ 2 TAB 24,CHR$ 1
 205 REM ----------------------
 210 PRINT TAB 10,CHR$ 3.CHR$ 3,
CHR$ 131,CHR$ 131,CHR$ 130
 215 REM ----------------------
 500 PRINT AT 1,0,"[redacted] FRANCE",TA
B 22,"GEOGRAPHIE",TAB 22,"======
====="
1000 PRINT AT 21,0,"APPUYER SUR
UNE TOUCHE"
1005 IF INKEY$="" THEN GOTO 1005
1010 LET A$="*****************
***************BONJOUR*********
***************VOICI LA CARTE O
E FRANCE.UN POINT VA CLIGNOTER S
UR L ECRAN ET 2 NOMS DE VILLES V
ONT APPARAITRE EN BAS DE L IMAGE
 AVEC CHACUN UN NUMERO,REPONDEZ
PAR LE NUMERO CHOISI ET L ORDINA
TEUR CONTROLERA VOTRE REPONSE.,.
,.,.,.,.,.ATTENTION,ETES-VOUS PRE
T ? (OUI/NON).....,"
1015 FOR N=1 TO LEN A$-31
1020 PRINT AT 21,0,A$(N TO N+31)
1025 FOR P=1 TO 2
1030 NEXT P
1035 NEXT N
1060 INPUT Z$
```

Vous pouvez aussi imaginer de transposer le programme pour un autre pays, Etats-Unis, Kenya, Ethiopie..., pour le plus grand plaisir des aventureux .

Vous constaterez peut-être alors que le score réalisé par un adulte sera bien différent du score qu'il aura obtenu avec la carte de France !

Utilisez la mémoire visuelle de vos enfants

Les lignes 10 à 210 permettent de dessiner le contour de la France Chaque ligne de PRINT correspond à l'impression d'une des vingt et une lignes de l'écran, afin de diminuer les risques de mauvaise interprétation des caractères à imprimer Cependant, disposant de la liste, vous pouvez, au lieu d'écrire par exemple $3, mettre son équivalent graphique, c'est-à-dire " [redacted] ", ce qui accélèrera l'affichage et diminuera l'encombrement en mémoire De même vous pouvez supprimer les REM qui ne sont là que pour augmenter la lisibilité du programme

La ligne 500 donne le titre du jeu et les lignes 1000 à 1005 invitent à jouer Quant aux lignes 1010 à 1035, elles initialisent dans la variable A$ la présentation du jeu et font défiler, sur la ligne 21, ce texte de droite à gauche en s'arrêtant sur la question « êtes-vous prêts ? » (oui/non). La boule P règle ce défilement.

Aux lignes 1060 à 1071, la question est posée et contrôlée

Les lignes 1075 à 1235 définissent le dimensionnement d'un tableau F$, dont chaque élément contiendra le nom d'une ville. L'originalité est la présence, dans les quatre premiers caractères du groupe, des deux numéros de deux chiffres correspondant

▶▶▶

à la position du plot de la ville.

Le ligne 1245 initialise le score S et la ligne 1250 demande le boucle N pour vingt questions.

Puis les lignes 1255 à 1280 calculant le numéro de deux villes, les coordonnées x et y de la première et le positionnent sur le carte.

Les lignes 1282 à 1287 donnent l'ordre d'écriture du nom de le ville.

Aux lignes 1290 à 1305, la boucle P fait clignoter le plot.

Les lignes 1310 à 1335 demandant la réponse, la contrôlent et affichent « faux » ou « correct », puis les lignes 1340 à 1345 calculent et affichent la score

Dans le cas d'une mauvaise réponse, le ligne 1360 donne la bonne Les lignes 1372 à 1375 effacent la plot et autorisent le retour à la ligne 1250.

Les lignes 1380 à 1390 donnent un commentaire sur la score, puis les lignes 1405 à 1420 invitent à redémarrer le jeu.

Remarques : les boucles P (lignes 1350-1355, 1365-1370 et 1395-1400) déterminent le temps de pause.

Le nombre de villes est ici de trente-deux, mais il est possible d'avoir plus de noms, en n'oubliant pas de modifier la valeur trente-deux des lignes 1075, 1255 et 1260, et en déplaçant le reste du programme.

Développer les connaissances de vos enfants en utilisant au mieux leur mémoire visuelle, tout en leur permettant une interaction, est une grande voie de l'informatique qui se développera intensivement dans les années à venir. Et pourquoi pas pour vous ?

René Lagache

▶▶▶

Programme de géographie (fin)

```
1065 IF Z$(1)="O" THEN GOTO 1075
1070 IF Z$(1)="N" THEN GOTO 1010
1071 GOTO 1060
1075 DIM F$(32,15)
1080 LET F$(1)="2533PARIS"
1085 LET F$(2)="2235ROUEN"
1090 LET F$(3)="2739ARRAS"
1095 LET F$(4)="1330RENNES"
1100 LET F$(5)="1425NANTES"
1105 LET F$(6)="1716BORDEAUX"
1110 LET F$(7)="2307TOULOUSE"
1115 LET F$(8)="0831BREST"
1120 LET F$(9)="3417LYON"
1125 LET F$(10)="3707MARSEILLE"
1130 LET F$(11)="4109NICE"
1135 LET F$(12)="3026BESANCON"
1140 LET F$(13)="4232STRASBOURG"
1145 LET F$(14)="1906TARBES"
1150 LET F$(15)="3114LE PUY"
1155 LET F$(16)="2425BOURGES"
1160 LET F$(17)="1927TOURS"
1165 LET F$(18)="2035LE HAVRE"
1170 LET F$(19)="3717CHAMBERY"
1175 LET F$(20)="1928LE MANS"
1180 LET F$(21)="3425DIJON"
1185 LET F$(22)="3134REIMS"
1190 LET F$(23)="3734METZ"
1195 LET F$(24)="1028VANNES"
1200 LET F$(25)="1619LA ROCHELLE
"
1205 LET F$(26)="1922POITIERS"
1210 LET F$(27)="3831NANCY"
1215 LET F$(28)="1726ANGERS"
1220 LET F$(29)="2331CHARTRES"
1225 LET F$(30)="2310LIMOGES"
1230 LET F$(31)="4708AJACCIO"
1235 LET F$(32)="4610BASTIA"
1245 LET S=0
1250 FOR N=1 TO 20
1255 LET V=INT (RND*32)+1
1260 LET T=INT (RND*32)+1
1265 IF T=V THEN GOTO 1260
1270 LET X=VAL F$(V,1 TO 2)
1275 LET Y=VAL F$(V,3 TO 4)
1280 PLOT X,Y
1281 PRINT AT 21,0;"
                    "
1282 LET H=INT (RND*2)+1
1283 IF H=1 THEN GOTO 1287
1285 PRINT AT 21,0,V,"-",F$(V,5
TO ),T,"-",F$(T,5 TO )
1286 GOTO 1290
1287 PRINT AT 21,0,T,"-",F$(T,5
TO ),V,"-",F$(V,5 TO )
1290 FOR P=1 TO 30
1295 UNPLOT X,Y
1300 PLOT X,Y
1305 NEXT P
1310 PRINT AT 3,24,"REPONSE?"
1315 INPUT Z$
1320 IF CODE Z$<29 OR CODE Z$>37
 THEN GOTO 1315
1325 LET Z=VAL Z$
1330 IF Z<>V THEN PRINT AT 3,24,
"[redacted]"
1335 IF Z=V THEN PRINT AT 3,24,"
[redacted]"
1340 IF Z=V THEN LET S=S+1
1345 PRINT AT 6,25,"SCORE ";AT 8
,25,S,"/",N
1350 FOR P=1 TO 10
1355 NEXT P
1360 IF Z<>V THEN PRINT AT 21,0,
"REPONSE JUSTE ",V,"-",F$(V,5 TO
 )
1365 FOR P=1 TO 20
1370 NEXT P
1372 UNPLOT X,Y
1373 PRINT AT 3,24,"          "
1375 NEXT N
1380 IF S<8 THEN PRINT AT 21,0;"
CE N EST PAS TRES BON,REVISEZ
"
1385 IF S>8 AND S<15 THEN PRINT
AT 21,0,"C EST UN SCORE MOYEN
                    "
1390 IF S>15 AND S<20 THEN PRINT
 AT 21,0,"C EST UN BON SCORE
                    "
1395 FOR P=1 TO 30
1400 NEXT P
1405 PRINT AT 21,0;"APPUYEZ SUR
UNE TOUCHE SVP      "
1410 IF INKEY$="" THEN GOTO 1410
1415 CLS
1420 RUN
```

vos chèques s'entrechoquent ? il faut les contrôler

Vous est-il déjà arrivé de vous lancer dans l'addition de, mettons, 500 chèques ? A votre gauche, les 500 chèques, au milieu une feuille de papier et un stylo, à votre droite, votre fidèle calculatrice de poche. Vous êtes calme, frais, dispos, et vous commencez à entrer vos premiers chiffres au clavier de la calculatrice. Ah ! Déjà une faute de frappe, bien ! Touche de correction d'erreur et l'on repart. A moins que... à moins que l'on ait tout effacé, auquel cas il faut repartir au premier chèque...

S'il vous reste quelques cheveux, c'est que vous n'avez pas souvent tenté cette expérience. L'essayer, c'est certainement ne pas l'adopter. De façon à vous aider à conserver votre crinière, voici la liste d'un programme BASIC, écrit à l'origine sur un TRS-80, programme qui vous aidera à surmonter plus facilement ces problèmes.

Première parution : L'OI n° 20

Le programme vous permettra de saisir un ensemble de chèques, arbitrairement limité à 1500, dimension du tableau T qui les contient.

Ces chèques pourront être imprimés en partie ou globalement et on pourra obtenir le total de tous les chèques entrés ou un sous-total quelconque entre deux limites choisies par l'opérateur.

Le problème des fautes de frappe est résolu de deux manières différentes : on teste la numéricité des zones qui doivent être numériques, ce qui permet de n'obtenir que des montants plausibles. Et si par malchance un montant se révèle faux lorsqu'on le visualise, il est possible de le modifier et de le remplacer par la valeur désirée.

La présentation se fait par pages numérotées (variable PA) entre une première ligne (PL) et une dernière ligne (DL) données.

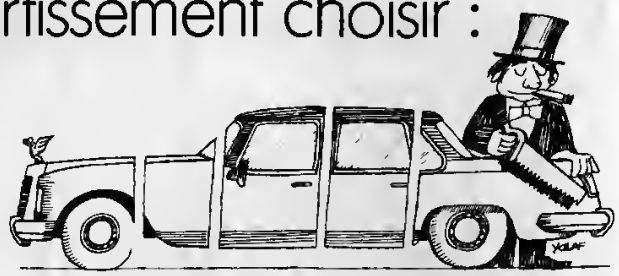
LT est la dernière ligne entrée dans le tableau, Z$ la variable d'entrée des montants et des choix d'aiguillage du programme. TL représente le total des chèques entrés à un moment donné. Les autres variables sont soit évidentes, soit de simples variables de passage utilisées ponctuellement.

Le programme présenté est très simple et, dans sa forme actuelle, ce n'est après tout qu'une « grosse » calculatrice. Il ne vous reste donc plus qu'à le développer en rajoutant par exemple une mémorisation sur fichier, puis la saisie de diverses zones comme le numéro du chèque, la date d'émission, la date de retrait, un code indiquant s'il a été ou non listé sur votre extrait de compte, tous ces petits détails qui le rendront propre à satisfaire exactement votre problème.

Jean-Pierre Madelt


```
10 REM**********************************************************
20 REM*          TRAITEMENT DE CHEQUES                         *
30 REM**********************************************************
40 REM
50 REM
60 REM          AUTEUR : J.P. MIGAUD
70 REM          CONCEPTION L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
80 REM**********************************************************
90 REM
100 REM         LISTE DES VARIABLES UTILISEES
110 REM
120 REM         VDM DU TABLEAU
130 REM         ZS          COMMANDE A EXECUTER OU VALEUR A ENTRER
140 REM         T(I)        TABLEAU DES VALEURS DES CHEQUES
150 REM         LT          NUMERO DE LA LIGNE COURANTE DU TABLEAU
160 REM         TL          TOTAL DU TABLEAU
170 REM         S           INDICATEUR D'ERREUR EN ENTREE
180 REM         PL          PREMIERE LIGNE D'EDITION
190 REM         DL          DERNIERE LIGNE D'EDITION
200 REM         PA          NUMERO DE PAGE DU TABLEAU
210 REM         AF          NUMERO DE LA PAGE A AFFICHER
220 REM         LN          COMPTAGE DES LIGNES D'UNE PAGE
230 REM**********************************************************
240 REM         PROGRAMME PRINCIPAL
250 REM
260 CLS
270 GOSUB 2000:         REM AFFICHAGE DU MODE D'EMPLOI
280 REM ON ATTEND UNE COMMANDE ET ON LA TRAITE
290 INPUT
300 IF ZS="C" THEN GOSUB 3000 GOTO 540  :REM CREATION DU TABLEAU
310 IF ZS="I" THEN GOSUB 3000 GOTO 540  :REM AFFICHAGE DU MODE
320 IF ZS="T" THEN GOSUB 4000 GOTO 540  :REM AFFICHAGE DU TOTAL GENERAL
330 IF ZS="P" THEN GOSUB 5000 GOTO 540  :REM AFFICHAGE D'UN TOTAL PARTIEL
340 IF ZS="I" THEN GOSUB 6000 GOTO 540  :REM IMPRESSION DU TABLEAU
350 IF ZS="L" THEN GOSUB 7000 GOTO 540  :REM IMPRESSION PARTIELLE DU TABLEAU
360 IF ZS="M" THEN GOSUB 8000 GOTO 540  :REM MODIFICATION D'UNE LIGNE
370 IF ZS="S" THEN GOTO 540             :REM POUR ELIMINER LES FAUTES DE FRAPPE
380 REM**********************************************************
390 CLS
400 PRINT                               :REM AFFICHAGE DU MODE D'EMPLOI
410 PRINT"                              MODE D'EMPLOI : PRINT.PRINT
420 PRINT" C    POUR COMMENCER UN TRAITEMENT"
430 PRINT" I    POUR AFFICHER LE MODE"
440 PRINT" T    POUR AVOIR UN TOTAL GENERAL DU TABLEAU"
450 PRINT" P    POUR AVOIR UN TOTAL PARTIEL DU TABLEAU"
460 PRINT" I    POUR IMPRIMER TOUT LE TABLEAU"
470 PRINT" L    POUR IMPRIMER UNE PARTIE DU TABLEAU"
480 PRINT" M    POUR MODIFIER UNE LIGNE DU TABLEAU"
490 PRINT" S    POUR SORTIR DU PROGRAMME" PRINT.PRINT
500 REM**********************************************************
510 REM         CREATION DU TABLEAU
520 REM**********************************************************
530 CLS  : DIM T(1500) : LT=1 : TL=0
540 INPUT "VERIFIER LA TITRE DU TABLEAU ";ZS
550 PRINT:PRINT"REPONDRE 'O' POUR CONTINUER, TAPER RETURN.";Z$
560 IF Z$="O" THEN GOTO 3220  :REM FIN DE CREATION DU TABLEAU
570 IF Z$="M" THEN GOSUB 8000 GOTO 3060 :REM CORRECTION D'ERREUR
580 REM DETECTION DE FAUTES DE FRAPPE (ON CONTROLE SI IL N'Y A QUE DES
590 REM CHIFFRES ; ON UN POINT DANS LA VALEUR ENTREE)
600 FOR I=1 TO LEN(Z$)
610         A$=MID$(Z$,I,1)
620         IF A$<"0" AND A$<> "." THEN PRINT"VOUS FAITES ERREUR" :GOTO 540
630         NEXT I
640 IF A$=" " THEN GOTO 3050
650 REM TOUT VA BIEN, ON INTRODUIT LA VALEUR DANS LE TABLEAU
660 T(LT)=VAL(Z$)
670 REM ET ON L'AJOUTE AU TOTAL GENERAL
680 TL=TL+T(LT)
690 REM PUIS ON PASSE A LA SUITE DE LA CREATION
700 LT=LT+1
710 GOTO 3050
720 RETURN
730 REM**********************************************************
740 REM         AFFICHAGE DU TOTAL GENERAL
750 REM**********************************************************
760 PRINT "TOTAL DU TABLEAU ";TL
770 REM**********************************************************
780 REM         AFFICHAGE D'UN TOTAL PARTIEL
790 INPUT "QUEL EST LE NUMERO DE LA PREMIERE LIGNE ";PL
800 INPUT "QUEL EST LE NUMERO DE LA DERNIERE LIGNE ";DL
810 FOR I=PL TO DL
820         TP=TP+T(I)
830 NEXT I
840 PRINT:PRINT TOTAL DES LIGNES "PL" A "DL" = ";TP
850 RETURN
860 REM**********************************************************
870 REM         IMPRESSION DU TABLEAU
880 REM**********************************************************
890 PL=1 DL=LT
900 GOSUB 7500
910 RETURN
920 REM**********************************************************
930 REM         IMPRESSION PARTIELLE DU TABLEAU
940 INPUT "PREMIERE LIGNE ";PL
950 INPUT"DERNIERE LIGNE ";DL
960 GOSUB 7500
970 RETURN
980 REM**********************************************************
990 REM         MODIFICATION
1000 INPUT NUMERO DE LIGNE A MODIFIER ";K
1010 PRINT"ANCIENNE VALEUR ";T(K)
1020 INPUT"QUELLE EST LA NOUVELLE VALEUR ";L
1030 T(K)=L
1040 RETURN
1050 REM**********************************************************
1060 REM         CHAGEMENT DE PAGE
1070 PRINT:PRINT"REPONDRE 'O' POUR CONTINUER, TAPER RETURN.";Z$
1080 CLS : LN=0
1090 PRINT, TAB(40) "TIS, PAGE ";PA PRINT
1100 REM CHIFFRES ; ON UN POINT DANS LA VALEUR ENTREE)
1110 FOR I=1 TO LEN(Z$)
1120         A$=MID$(Z$,I,1)
1130         IF A$<"0" AND A$> "9" THEN PRINT VOUS FAITES ERREUR" :GOTO 540
1140 NEXT I
1150 IF A$=" " THEN GOTO 3050
1160 REM TOUT VA BIEN, ON INTRODUIT LA VALEUR DANS LE TABLEAU
1170 T(LT)=VAL(Z$)
1180 REM ET ON L'AJOUTE AU TOTAL GENERAL
1190 TL=TL+T(LT)
1200 REM PUIS ON PASSE A LA SUITE DE LA CREATION
1210 LT=LT+1
1220 GOTO 3050
1230 RETURN
1240 REM**********************************************************
1250 REM         AFFICHAGE DU TOTAL GENERAL
1260 REM**********************************************************
1270 CLS : PRINT "TOTAL DU TABLEAU ";TL
1280 REM**********************************************************
1290 REM         AFFICHAGE D'UN TOTAL PARTIEL
1300 INPUT "QUEL EST LE NUMERO DE LA PREMIERE LIGNE ";PL
1310 INPUT "QUEL EST LE NUMERO DE LA DERNIERE LIGNE ";DL
1320 FOR I=PL TO DL
1330         TP=TP+T(I)
1340 NEXT I
1350 PRINT:PRINT TOTAL DES LIGNES "PL" A "DL" = ";TP
1360 RETURN
1370 REM**********************************************************
1380 REM         IMPRESSION DU TABLEAU
1390 REM**********************************************************
1400 PL=1 DL=LT
1410 GOSUB 7500
1420 RETURN
1430 REM**********************************************************
1440 REM         IMPRESSION PARTIELLE DU TABLEAU
1450 INPUT "PREMIERE LIGNE ";PL
1460 INPUT"DERNIERE LIGNE ";DL
1470 GOSUB 7500
1480 RETURN
1490 REM**********************************************************
1500 REM         MODIFICATION
1510 INPUT NUMERO DE LIGNE A MODIFIER ";K
1520 PRINT"ANCIENNE VALEUR ";T(K)
1530 INPUT"QUELLE EST LA NOUVELLE VALEUR ";L
1540 T(K)=L
1550 RETURN
1560 REM**********************************************************
1570 REM         CHAGEMENT DE PAGE
1580 PRINT:PRINT"REPONDRE 'O' POUR CONTINUER, TAPER RETURN.";Z$
1590 CLS : LN=0
1600 PRINT, TAB(40) "TIS, PAGE ";PA PRINT
1610 PL=1 DL=LT
1620 GOSUB 7500
1630 RETURN
1640 REM**********************************************************
1650 REM         IMPRESSION PARTIELLE DU TABLEAU
1660 INPUT "PREMIERE LIGNE ";PL
1670 INPUT"DERNIERE LIGNE ";DL
1680 GOSUB 7500
1690 RETURN
1700 REM**********************************************************
1710 REM         MODIFICATION
1720 INPUT NUMERO DE LIGNE A MODIFIER ";K
1730 PRINT"ANCIENNE VALEUR ";T(K)
1740 INPUT"QUELLE EST LA NOUVELLE VALEUR ";L
1750 T(K)=L
1760 RETURN
1770 REM**********************************************************
1780 REM         CHAGEMENT DE PAGE
1790 PRINT:PRINT"REPONDRE 'O' POUR CONTINUER, TAPER RETURN.";Z$
1800 CLS : LN=0
1810 PRINT, TAB(40) "TIS, PAGE ";PA PRINT
1820 PL=1 DL=LT
1830 GOSUB 7500
1840 RETURN
1850 REM**********************************************************
1860 REM         IMPRESSION PARTIELLE DU TABLEAU
1870 INPUT "PREMIERE LIGNE ";PL
1880 INPUT"DERNIERE LIGNE ";DL
1890 GOSUB 7500
1900 RETURN
1910 REM**********************************************************
1920 REM         MODIFICATION
1930 INPUT NUMERO DE LIGNE A MODIFIER ";K
1940 PRINT"ANCIENNE VALEUR ";T(K)
1950 INPUT"QUELLE EST LA NOUVELLE VALEUR ";L
1960 T(K)=L
1970 RETURN
1980 REM**********************************************************
1990 REM         CHAGEMENT DE PAGE
2000 PRINT:PRINT"REPONDRE 'O' POUR CONTINUER, TAPER RETURN.";Z$
2010 CLS : LN=0
2020 PRINT, TAB(40) "TIS, PAGE ";PA PRINT
2030 PL=1 DL=LT
2040 GOSUB 7500
2050 REM**********************************************************
2060 REM         IMPRESSION PARTIELLE DU TABLEAU
2070 INPUT "PREMIERE LIGNE ";PL
2080 INPUT"DERNIERE LIGNE ";DL
2090 GOSUB 7500
2100 RETURN
2110 REM**********************************************************
2120 REM         MODIFICATION
2130 INPUT NUMERO DE LIGNE A MODIFIER ";K
2140 PRINT"ANCIENNE VALEUR ";T(K)
2150 INPUT"QUELLE EST LA NOUVELLE VALEUR ";L
2160 T(K)=L
2170 RETURN
2180 REM**********************************************************
2190 REM         CHAGEMENT DE PAGE
2200 PRINT:PRINT"REPONDRE 'O' POUR CONTINUER, TAPER RETURN.";Z$
2210 CLS : LN=0
2220 PRINT, TAB(40) "TIS, PAGE ";PA PRINT
2230 PL=1 DL=LT
2240 GOSUB 7500
2250 REM**********************************************************
2260 REM         IMPRESSION PARTIELLE DU TABLEAU
2270 INPUT "PREMIERE LIGNE ";PL
2280 INPUT"DERNIERE LIGNE ";DL
2290 GOSUB 7500
2300 RETURN
2310 REM**********************************************************
2320 REM         MODIFICATION
2330 INPUT NUMERO DE LIGNE A MODIFIER ";K
2340 PRINT"ANCIENNE VALEUR ";T(K)
2350 INPUT"QUELLE EST LA NOUVELLE VALEUR ";L
2360 T(K)=L
2370 RETURN
2380 REM**********************************************************
2390 REM         CHAGEMENT DE PAGE
2400 PRINT:PRINT"REPONDRE 'O' POUR CONTINUER, TAPER RETURN.";Z$
2410 CLS : LN=0
2420 PRINT, TAB(40) "TIS, PAGE ";PA PRINT
2430 PL=1 DL=LT
2440 GOSUB 7500
2450 REM**********************************************************
2460 REM         IMPRESSION PARTIELLE DU TABLEAU
2470 INPUT "PREMIERE LIGNE ";PL
2480 INPUT"DERNIERE LIGNE ";DL
2490 GOSUB 7500
2500 RETURN
2510 REM**********************************************************
2520 REM         MODIFICATION
2530 INPUT NUMERO DE LIGNE A MODIFIER ";K
2540 PRINT"ANCIENNE VALEUR ";T(K)
2550 INPUT"QUELLE EST LA NOUVELLE VALEUR ";L
2560 T(K)=L
2570 RETURN
2580 REM**********************************************************
2590 REM         CHAGEMENT DE PAGE
2600 PRINT:PRINT"REPONDRE 'O' POUR CONTINUER, TAPER RETURN.";Z$
2610 CLS : LN=0
2620 PRINT, TAB(40) "TIS, PAGE ";PA PRINT
2630 PL=1 DL=LT
2640 GOSUB 7500
2650 REM**********************************************************
2660 REM         IMPRESSION PARTIELLE DU TABLEAU
2670 INPUT "PREMIERE LIGNE ";PL
2680 INPUT"DERNIERE LIGNE ";DL
2690 GOSUB 7500
2700 RETURN
2710 REM**********************************************************
2720 REM         MODIFICATION
2730 INPUT NUMERO DE LIGNE A MODIFIER ";K
2740 PRINT"ANCIENNE VALEUR ";T(K)
2750 INPUT"QUELLE EST LA NOUVELLE VALEUR ";L
2760 T(K)=L
2770 RETURN
2780 REM**********************************************************
2790 REM         CHAGEMENT DE PAGE
2800 PRINT:PRINT"REPONDRE 'O' POUR CONTINUER, TAPER RETURN.";Z$
2810 CLS : LN=0
2820 PRINT, TAB(40) "TIS, PAGE ";PA PRINT
2830 PL=1 DL=LT
2840 GOSUB 7500
2850 REM**********************************************************
2860 REM         IMPRESSION PARTIELLE DU TABLEAU
2870 INPUT "PREMIERE LIGNE ";PL
2880 INPUT"DERNIERE LIGNE ";DL
2890 GOSUB 7500
2900 RETURN
2910 REM**********************************************************
2920 REM         MODIFICATION
2930 INPUT NUMERO DE LIGNE A MODIFIER ";K
2940 PRINT"ANCIENNE VALEUR ";T(K)
2950 INPUT"QUELLE EST LA NOUVELLE VALEUR ";L
2960 T(K)=L
2970 RETURN
2980 REM**********************************************************
2990 REM         CHAGEMENT DE PAGE
3000 PRINT:PRINT"REPONDRE 'O' POUR CONTINUER, TAPER RETURN.";Z$
3010 CLS : LN=0
3020 PRINT, TAB(40) "TIS, PAGE ";PA PRINT
3030 PL=1 DL=LT
3040 GOSUB 7500
3050 REM**********************************************************
3060 REM         IMPRESSION PARTIELLE DU TABLEAU
3070 INPUT "PREMIERE LIGNE ";PL
3080 INPUT"DERNIERE LIGNE ";DL
3090 GOSUB 7500
3100 REM CHIFFRES ; ON UN POINT DANS LA VALEUR ENTREE)
3110 FOR I=1 TO LEN(Z$)
3120         A$=MID$(Z$,I,1)
3130         IF A$<"0" AND A$> "9" THEN PRINT VOUS FAITES ERREUR" :GOTO 540
3140 NEXT I
3145 IF A$=" " THEN GOTO 3050
3150 REM TOUT VA BIEN, ON INTRODUIT LA VALEUR DANS LE TABLEAU
3160 T(LT)=VAL(Z$)
3170 REM ET ON L'AJOUTE AU TOTAL GENERAL
3180 TL=TL+T(LT)
3190 REM PUIS ON PASSE A LA SUITE DE LA CREATION
3200 LT=LT+1
3210 GOTO 3050
3220 RETURN
4000 REM**********************************************************
4010 CLS : PRINT "TOTAL DU TABLEAU ";TL
4020 RETURN
5000 REM**********************************************************
5005 TP=0
5010 INPUT "QUEL EST LE NUMERO DE LA PREMIERE LIGNE ";PL
5015 INPUT "QUEL EST LE NUMERO DE LA DERNIERE LIGNE ";DL
5020 FOR I=PL TO DL
5040 TP=TP+T(I)
5045 NEXT I
5050 PRINT:PRINT TOTAL DES LIGNES "PL" A "DL" = ";TP
5060 RETURN
6000 REM**********************************************************
6010 PL=1 DL=LT
6020 GOSUB 7500
6030 RETURN
6040 REM**********************************************************
6050 REM         IMPRESSION PARTIELLE DU TABLEAU
6060 INPUT "PREMIERE LIGNE ";PL
6070 INPUT"DERNIERE LIGNE ";DL
7000 REM**********************************************************
7010 REM         IMPRESSION DE PL A DL
7020 PA=1
7030 GOSUB 7500
7040 RETURN
7500 REM**********************************************************
7510 PA=1
7520 GOSUB 9000 :REM IMPRESSION DE TETE DE PAGE
7530 FOR I=PL TO DL
7540         IF LN=15 THEN PA=PA+1 :GOSUB 9000
7550         LN=LN+1
7560         PRINT I,T(I)
7570 NEXT I
7580 RETURN
8000 REM**********************************************************
8010 INPUT NUMERO DE LIGNE A MODIFIER ";K
8020 PRINT"ANCIENNE VALEUR ";T(K)
8030 INPUT"QUELLE EST LA NOUVELLE VALEUR ";L
8040 T(K)=L
8050 T(K)=L
8060 RETURN
9000 REM**********************************************************
9010 REM         CHAGEMENT DE PAGE
9020 CLS : LN=0
9030 PRINT, TAB(40) "TIS, PAGE ";PA PRINT
9040 PRINT"LIGNE ","VALEUR"
9050 RETURN
```

quel amortissement choisir : linéaire ou dégressif



L'amortissement fait partie du vocabulaire couramment utilisé par la plupart des entreprises.« Les mesures fiscales pour favoriser l'investissement » ne sont rien d'autre que des lois qui assouplissent ou restreignent les procédures d'amortissement. Alors que vous préfériez l'amortissement linéaire ou l'amortissement dégressif, pourquoi ne pas utiliser le petit programme Apple que nous vous présentons.

La notion d'amortissement est intimement liée à celle d'investissement. Pourtant, dans la vie de tous les jours, nous pensons beaucoup aux investissements que nous aimerions faire – nouvelle voiture, appartement, meubles, etc. – et assez peu aux amortissements.

Bien sûr, s'il vous arrive de calculer le prix de revient de votre voiture par mois ou par kilomètre, vous y incluez sans doute une rubrique que vous appelez pompeusement « amortissement », à moins que vous ne préfériez « usure » ou « dépréciation de la voiture ». La signification que vous donnez à ces mots n'est peut-pas très précise, mais vous sentez que l'argent avec lequel vous avez payé cette voiture n'a pas été totalement « dépensé » le jour où vous avez fait la chèque. Cette grosse dépense, cet investissement doit être étalé sur une certaine durée. C'est d'ailleurs ce que nous faisons tous quand nous achetons à crédit, et même quand nous mettons de l'argent de côté en prévision d'un achat.

Mais c'est dans les entreprises que l'utilisation des amortissements a trouvé son terrain d'élection, tout simplemant parce qu'elles tiennent une comptabilité précise selon un schéma figé. Pourquoi et comment est établie cette comptabilité, c'est ce que nous allons voir ensemble.

On peut faire ses comptes de multiples façons, qui toutes se ramènent à deux possibilités. Dans la première, vous mettez dans une colonne ce que vous gagnez, pendant une certaine période, dans l'autre vos dépenses pendant la même période. La différence entre les deux colonnes est un déficit que vous devrez puiser dans vos réserves ou un excédant que vous pouvez mettre de côté – à moins que vous ne préfériez vous offrir quelque chose qui vous fait envie.

L'autre possibilité est de mettre d'un côté ce que vous possédez, de l'autre ce que vous devez. La différence entre les deux vous donne la valeur de votre patrimoine à un moment donné, qui peut être positif ou négatif.

Pour le commun des mortels, la première méthode est pratiquement la seule utilisée, car nous vivons au jour le jour avec des besoins limités en matière de prévision et de précision. La grande question est en général : « Est-ce que je peux me permettre de... ? » et nous ne cherchons pas toujours à ce que la réponse soit fondée uniquement sur des arguments rationels.

L'entreprise, par contre, utilise en même temps ces deux types de comptes, dans un ensemble qui porte le nom de comptabilité générale.

Tous les ans, l'entreprise établit son Bilan, qui est une photographie du patrimoine de l'entreprise. Dans la colonne de gauche figurent les Actifs, c'est-à-dire ce que l'entreprise possède, en particulier les Immobilisations qui correspondent aux investissements de l'entreprise), les créances et les liquidités. Rappelons qu'une créance est le contraire d'une dette : c'est la marque que quelqu'un vous doit de l'argent.

Dans la colonne de droite, le Passif, on trouve le Capital et les Réserves, les Dettes et le Résultat ou Bénéfice (s'il y en a). C'est un assemblage un peu étrange, car on ne voit pas spontanément de rapport entre le capital et les dettes d'une entreprise. Il y a pourtant deux points communs. Le capital est en quelque sorte une dette envers les actionnaires qui ont investi dans l'entreprise, et qu'ils doivent récupérer à la dissolution de celle-ci (voir tableau 1).

Cette façon de voir est un peu artificielle, mais la second point commun est beaucoup plus fort. En effet capital et dettes correspondent à de l'argent qui est entré dans l'entreprise. Cet argent a pu servir à acheter des machines ou à renflouer la trésorerie, donc à créer des sommes qui figurent à l'Actif du Bilan. De cette façon, capital et dettes jouent le même rôle dans le système comptable. Quant aux réserves, c'est essentiellement ce qui reste des bénéfices des années précédentes après distribution des dividendes et paiement des impôts.

Ne vivez plus au jour le jour

Entre deux bilans s'écoule une période, généralement d'un an, que l'on appelle Exercice Comptable. Pendant cette période, l'entreprise travaille avec le Compte d'Exploitation (ou Compte de Résultats), qui est vide en début d'exercice, et se nourrit peu à peu de charges et de produits.

Les produits et les charges sont des avaters assez sophistiqués des colonnes de tout à l'heure : je gagne - je dépense. On y trouve en particulier les Amortissements dont nous allons reparler. La différence entre les produits et les charges est le Bénéfice, qui se trouve être le même que celui qui figurera au Bilan en fin d'exercice.

Tout ce système est très rigide car il intéresse énormément des personnes extérieures à l'entreprise. Pour des raisons juridiques, il faut savoir qui doit quoi à qui, et qui possède quoi ; c'est pour cela que le bilan annuel est un document public.

Mais surtout le fisc regarde toujours de très près comment est calculé le bénéfice, puisqu'il tient à en toucher la moitié au titre de l'impôt direct sur les sociétés. Et dites-vous bien que votre feuille de déclaration de revenus est une petite plaisanterie à côté de ce qu'une entreprise doit fournir à l'administration fiscale pour le calcul de ses impôts !

Première parution : L'OI n° 31

BILAN FIN 80	
ACTIF	PASSIF

BILAN FIN 81	
ACTIF	PASSIF
Investissements	Capital et réserves
Créances	Dettes
Liquidités	
	Bénéfice

CE 81	
CHARGES	PRODUITS
Salaires Achats Impôts Frais divers Amortissements	Ventes
Bénéfice	

Tableau 1

Jusqu'ici il n'y a pas vraiment de problème : le bénéfice est la différence entre les produits et les charges de l'exercice, et l'Etat en prend la moitié !

Mais si vous achetez une machine qui va vous durer 5 ans, les ennuis commencent. Si cet achat était tout de suite considéré comme une charge, l'exercice serait sans doute déficitaire, ce qui fait toujours mauvais effet, et le fisc ne toucherait rien et ne serait pas content. Les années suivantes, le bénéfice serait important puisque la machine serait utilisée sans rien coûter apparemment. A moins d'achater une nouvelle machine.

Dans cette situation où les résultats jouent aux montagnes russes et où les entreprises sont invitées à se surèquiper pour ne pas payer d'impôts, il serait très difficile d'apprécier la santé d'une entreprise.

C'est ici que les amortissements entrent en scène : ils permettent d'amortir le choc d'un investissement en diluant ses effets sur plusieurs années.

L'idée est très simple : l'achat d'une machine n'est pas une charge. Par exemple, si elle est payée au comptant, les liquidités vont diminuer et les immobilisations vont augmenter du même montant : aucune influence sur le bénéfice. Par contre, à la fin de chaque exercice, la valeur de la machine (sa valeur « comptable ») sera diminuée d'un certain montant, l'amortissement, qui figure, lui, en charge et donc diminue le bénéfice. Au bout de quelques exercices, la valeur comptable de la machine sera nulle et elle aura été complètement amortie.

Vous imaginez sans peine que la façon dont sont calculés les amortissements a une grosse influence sur les bénéfices, donc sur les impôts que payent les entreprises. Un exemple le montrera clairement.

Supposons qu'une entreprise achète une machine de 120 000 francs, pour une fabrication qui doit lui « rapporter » 35 000 francs par an pendant six ans, sans tenir compte de l'amortissement. Je n'insiste pas sur ce que veut dire exactement « rapporter » : disons que ces 35 000 francs sont une « marge ».

Le tableau ci-dessous vous montre comment seront calculés les impôts de l'entreprise avec trois modes de calcul des amortissements. Les deux premiers cas sont très simples : on amortit une somme égale chaque année, c'est ce que l'on appelle l'amortissement linéaire. Le troisième exemple correspond à l'amortissement dégressif, qui est un peu plus compliqué et dont je parlerai un peu plus loin.

Dans ces trois exemples, vous pouvez constater que les totaux des trois colonnes ne changent pas avec le mode de calcul. Le montant total d'impôt payé sur six ans est toujours le même. Mais dans les deux derniers cas, ces impôts sont payés plus tard, ce qui présente un intérêt évident, surtout en ces temps d'inflation importante.

Amortissement linéaire sur six ans

	Marge	Amortissement	Bénéfice	Impôt
1	35 000	20 000	15 000	7 500
2	35 000	20 000	15 000	7 500
3	35 000	20 000	15 000	7 500
6	35 000	20 000	15 000	7 500
	210 000	120 000	90 000	45 000

Amortissement linéaire sur trois ans

	Marge	Amortissement	Bénéfice	Impôt
1	35 000	40 000	– 5 000	0
2	35 000	40 000	– 5 000	0
3	35 000	40 000	– 5 000	0
4	35 000	0	35 000	10 000
5	35 000	0	35 000	17 500
6	35 000	0	35 000	17 600
	210 000	120 000	90 000	45 000

Amortissement dégressif sur six ans

1	35 000	40 000	– 5 000	0
2	35 000	26 667	8 333	1 666,50
3	35 000	17 778	17 222	8 611
4	35 000	11 852	23 148	11 574
5	35 000	11 852	23 148	11 574
6	35 000	11 851	23 149	11 574,50
	210 000	120 000	90 000	45 000

Tableau 2

Dans cas tableaux certains bénéfices sont négatifs, dans ca cas les impôts sont nuls (l'Etat subventionna parfois las entreprises an péril mais pas systématiquement !). Il y a alors report des pertas sur les années qui présantent un bénéfice, ca qui axplique que l'impôt est parfois inférieur à la moitié du bénéfice. Mais si l'antreprise avait d'autres activités bénéficiairas, la situation serait différente. Le déficit des premières années viandrait effectivement diminuer le bénéfice total de l'entreprisa, et donc l'impôt payé. Dans ce cas, le déplacemant des impôts vers l'avanir serait ancore plus grand.

Une exponentielle n'a pas de fin...

Ces exemples nous montrent que l'entreprise désire pouvoir amortir le plus vite possible ses investissements pour diminuer son bénéfice « fiscal », lequel n'a pas toujours une signification trés réaliste. Les deux moyens d'y parvenir sont de diminuer la durée d'amortissement ou de prendre un amortissement dégressif. Inutile de vous dire que le Code des Impôts est très précis sur ce que l'on a le droit de faire.

Puisque c'est si intéressant, comment donc est calculé l'amortissement dégressif ? Le principe ast d'amortir chaque année une somme proportionnelle à la valeur comptable, c'ast-á-dire à ce qui reste après avoir enlevé les amortissements déjà réalisés. Sur le graphique 3 la valeur comptable de l'investissement décrit donc une courbe exponentielle, au lieu d'une droite comme dans le cas de l'amortissement linéaire. Mais une exponentielle n'a pas de fin, aussi, pour éviter que les choses ne traînent en longueur, on reprend un amortissement linéaire vers la fin de la durée de vie de la machine (dans notre exemple, sur les deux dernières années).

Le rythme d'amortissement est donné par ce que l'on appelle taux d'amortissement. Pour l'amortissement linéaire, le taux est simplement exprimé en % en divisant 100 par la durée de vie du bien. Pour l'amortissement dégressif, on calcule d'abord le taux constant comme dans le cas linéaire, puis on le multiplie par un coefficient généralement compris entre 1,5 et 3, pour obtenir le taux dit dégressif. Actuallement en Franca, le coefficient multiplicateur dépand de la durée d'amortissement, de la façon suivante :

de 2 à 4 ans 1,5
jusqu'à 6 ans 2,0
supérieur à 6 ans 2,5

Dans notra exemple, sur six ans, le taux constant vaut 1/6 soit 16,67 % et le taux dégressif vaut 1/3 soit 33,33 %.

Nous avons parlé jusqu'ici des amortissements tels qu'ils sont pratiqués en comptabilité générale, la seule qui soit obligatoire et stricte pour toutes les entreprises. Les résultats auxquels nous sommes arrivés nous ont montré que ces calculs sont souvent artificiels. Si vous avez le droit d'utiliser l'amortissement dégressif, ça vous fera plaisir parce que vous payerez vos impôts plus tard, mais ça ne correspond á aucune réalité économique.

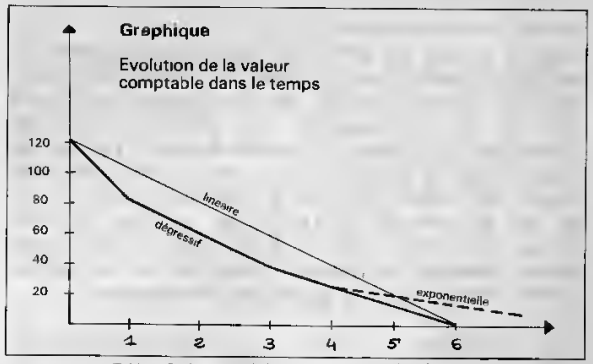


Tableau 3 : évolution de la valeur comptable dans le temps

Par contre, dans de nombreux cas, il est nécessaire de calculer des amortissements sur des bases différentes, pour obtenir des indications plus fiables sur, par exemple, la rentabilité d'un investissement. Le problème se pose en particulier pour le calcul des prix de revient – vous vous souvenez de l'exemple de la voiture au début de cet article ? Ceci explique que les entreprises calculent parfois l'amortissement de la même machine de trois ou quatre façons différentes selon l'usage qui en sera fait.

Le programme que nous vous proposons calcule les amortissements linéaires ou dégressifs et les présente sous forme de tableau, par année, avec la somme á amortir, et la valeur comptable an fin d'exercica (cella qui figurera au bilan en fin d'année). Il est écrit en Applesoft et ne présente pas grande difficulté ! Las données vous sont demandées de façon claire, il n'y a donc pas besoin d'instructions spécialas pour que vous puissiez l'utiliser. Le programme contrôle que toutes les données sont plausibles, en particulier qua la durée n'est pas fractionnaire (las demi-années sont accaptées) et qua la valeur ne comprend que des centimes (las millimes n'existent pas en France).

Dans tous les programmes da comptabilité ou de gestion, on trouve d'ailleurs ces problémes d'arrondi et de nombre de chiffres aprés la virgule. Les centimes doivent être exacts même s'ils n'ont aucune signification. Cas problémes sont ici résolus par l'emploi systématique des fonctions définies au début du programme : F2 (X) est l arrondi au centime le plus proche de X ; ND (X) est le nombre de chiffres après la virgule que possède X (attention : c'est – 1 si X est entier).

Il y a enfin un problème dont nous n avons pas parlé, celui de la première année. Quand un bien est acquis en cours d'année (c'est-à-dire dans la plupart des cas) l'amortissement de la première année est réduit á une partie da ce qu'il aurait été pour une année pleine. Ceci se fait proportionnellement au nombre de jours écoulés entre l'achat et la fin de l'année. Mais le calcul est simple, car il s'agit de jours « comptables », sur la base d'une année fictive de 12 mois tous égaux da 30 jours.

Guilhem Bouteloup

attendre... attendre... toujours attendre !

Que ce soient les artisans (faut-il embaucher ?), les industriels (faut-il faire un stock ou l'augmenter s'il existe déjà), les commerçants (faut-il agrandir la boutique ?), les grandes surfaces (combien de caisses enregistreuses à tel ou tel moment de la journée ?), les transports de marchandises (quel doit être le volume du camion ?) ou de personnes (attente au guichet SNCF, péage d'autoroute, etc.), bref, tout ce monde se trouve devant le même problème de file d'attente, mais avec des données différentes !

Une file d'attente est provoquée :

par l'arrivée d'individus à intervalles fixes ou variables d'une part ;

par la prise en charge ou le traitement de ces individus à intervalles fixes ou variables d'autre part.

Jusqu'ici les choses sont limpides mais, hélas, incomplètes. En effet, les arrivées ou les traitements peuvent être simples ou multiples et, dans le cas où ils sont multiples, le nombre d'individus lui aussi peut être fixe ou variable (ça se complique !).

Ce n'est pas tout : le fait d'admettre que des éléments sont variables implique que cette variabilité suive les lois mathématiques connues, de façon à pouvoir simuler les variations que l'on rencontre couramment dans la « nature » (employé ici au sens très large du terme !).

Les concepts statistiques viendront à notre secours en nous donnant des modèles mathématiques pouvant reconstituer à peu près tous les cas de figures en matière de variation.

L'ordinateur simule les variations rencontrées dans la nature

Trois types de variations sont mis en œuvre :

— aléatoire : chaque valeur délimitée par deux bornes (inférieure et supérieure) a la même probabilité d'être générée ;

— gaussien : la probabilité d'une valeur est d'autant plus forte qu'elle est située à proximité de la moyenne, elle-même centrée par rapport aux valeurs extrêmes ; les caractéristiques définissant cette loi sont la moyenne et l'écart type (variation des valeurs autour de la moyenne ± trois écarts types) ;

— poissonnien : analogue à la loi de Gauss, à la différence près que la moyenne peut être plus ou moins décentrée ; les caractéristiques définissant cette loi sont la moyenne, l'écart type et le centrage de la moyenne (six écarts types entre les valeurs extrêmes) ; ici, la loi de Poisson est utilisée de façon un peu inhabituelle mais adéquate (les contestataires sont priés de se faire connaître !).

Pour chacun de ces trois types, un petit programme (tout petit...) permet de vérifier que la simulation est fidèle au modèle de variation choisi. Pour ce faire, ces programmes sont construits autour d'un même schéma :

— génération des valeurs correspondant au modèle de variation,

— classification de ces valeurs par ordre croissant,

— regroupement par classe ayant chacune le même intervalle (exemple : 10 à 15, 15 à 20, 20 à 25...).

Le nombre de classes sera égal à la racine carrée du nombre de valeurs avec un plafond à 15 (c'est-à-dire un écran). (Ah oui ! j'oubliais de vous dire : ça tourne sur un TRS 80 modèle 1, 16 Ko garanti !)

Le tout est mis dans un superbe graphique avec les classes en ordonnées et les effectifs de classe en abcisses, et s'appelle « histogramme de fréquences ».

Ce programme est d'ailleurs inclus dans le programme principal pour visualiser la répartition, non plus des valeurs générées, mais des attentes d'arrivées ou de traitements (au choix).

Cela nous amène tout naturellement au fonctionnement du programme principal dans lequel on observe trois phases successives : l'entrée des données, la simulation, l'affichage des résultats.

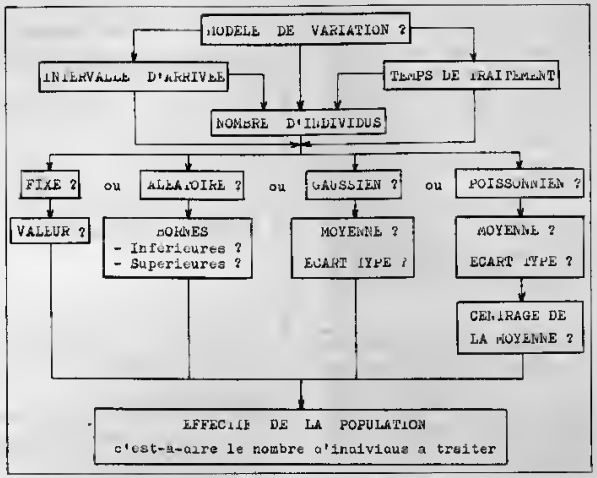
1) L'entrée des données : une foule de questions vous sont posées, mais aucune cependant n'est superflue.

En premier lieu, il vous est demandé quel type d'analyse vous voulez faire : les attentes de traitement ou les attentes d'arrivées, cela afin de ne stocker que les données nécessaires (16 Ko oblige !) ; cette analyse s'effectuera à l'aide de l'histogramme.

Une brève question ensuite sur les arrivées ou traitements simultanés ou non ; cela afin d'écourter le questionnaire dans le cas où les arrivées et les traitements sont simples.

Ensuite, l'ordinateur vous met en condition pour affronter le reste du questionnaire. Il rappelle en même temps que tout a une limite (et il fait partie du tout) et vous engage à ne pas dépasser les dosas prescrites afin de préserver son équilibre mental !

Vient alors le « rush » des questions embêtantes qui peuvent vous mettre dans l'embarras si vous n'avez pas effectué un minimum de travaux de préparation. On peut résumer ces questions suivant le schéma ci-après :



Il va de soi que, pour avoir une réponse à donner à toutes ces questions, il faut aller « à la pêche » (là aussi au sens très large du terme) ou faire preuve d'un grand talent d'improvisation. Si vous avez des relevés sur ce qui s'est déjà produit, le petit programme ci-dessous vous sera d'un grand secours pour définir vos modèles de variation.

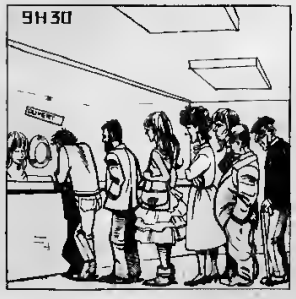
Eh bien voilà, c'est terminé pour la première phase (ouf !).

2) la simulation : là, deux situations possibles :

— le jeu des questions et des réponses vous a traumatisé, alors vous enfoncez la touche CLEAR et vous n'aurez pas besoin de vous voiler la face ; seuls deux petits chiffres apparaîtront au centre de

Tracé d'un histogramme avec valeurs en DATA

```
10 CLEAR:DIMF(1600),G(1:00):READA:MIN=A:PESTORE
.0 FORI=1TO1600:READA:IFA=0THENI=1600:GOTO50
30 H=H+1:A=INT(A):F(A)=F(A)+1:IFMIN<ATHENMIN=A
40 PRINTUSING" ###### ";A;:IFMAX<ATHENMAX=A
50 NEXTI:IFE<>R H) 15THEND=INT(SQR(H)+1)ELSED=15
60 M DY=MAX-MIN:C=INT(MDY/D+1):M=1
70 CLS:FORJ=INT MIN)TOINT MAX)STEPC:E=J+C
80 PRINTUSING"###";J;:PRINT' A ";USING"### ";E;
90 FORI=JTOJ+C-1:G(J)=G(J)+F(I):NEXTI
100 IFG(J)/M 47THENM=M+1:FORI=JTOMINSTEP-C:G(I)=0:NEXTI:GOTO70
110 IFG(J) 0THENFORI=1TOG(J),M:PRINTCHR$(143);:NEXTI
120 IFG J)=0THENPRINT:NEXTJ:GOTO140
170 PRINTINT(G(J H*1000+.5)/10;"%";:NEXTJ
140 IFPEEK(14400) 11THEN140ELSE10
```



l'écran : les nombres d'individus traités et en attente ;

– vous êtes sûr de la représentativité de vos réponses, alors vous enfoncez le touche ENTER et observez, étape par étape, le déroulement des événements (c'est-à-dire le détail de le file avec l'heure d'arrivée de cheque individu en attente).

Il est évidemment possible, dans le cas d'un brusque changement d'avis, de revenir à l'une ou l'autre situation en utilisent les touches adéquates (CLEAR ou ENTER)

Point n'est besoin d'imprimente pour visualiser le détail de la simulation. La gestion des tableaux est faite de façon que l'écran remplisse parfaitement son rôle d'économiseur de papier, d'autant plus que, comme toute simulation qui se respecte, elle doit être répétée un certain nombre de fois.

Une seule chose à faire : ne rien faire

Une recommandation : il peut arriver que votre TRS, pris de stupeur, soit frappé de parelysie le condamnant à une immobilisation complète (qui ne sere que temporaire, je vous rassure). Si cela venait à se produire, surtout pas de panique et dites-vous bien que vous n'en êtes pas responsable ! Oe plus, il n'y e qu'une seule chose à faire : ne rien faire !

Il s'egit, en feit, suite à un temps d'attente trop long pour la taille mémoire disponible, d'un changement d'adresse et d'un regroupement partiel de toutes les données déjà collectées de sorte que les adresses de stockage ne dépassent pas la réserve de mémoire prévue (ce sont, en feit, les temps d'attente qui servent d'edresses).

3) L'affichage des résultats : une première partie rappelle de façon condensée les données de départ avec, en plus, les ratios intervalle/nombre d'individus qui permettent de dégager la tendance des attentes (équilibre si les deux retios sont égaux, sinon les attentes augmentent là où le retio est le plus fort) ; cele seulement dans le cas d'arrivées ou traitements multiples.

Une seconde partie donne en valeur maximale et en moyenne :

GENERATION DE NOMBRES : - Modèle ALEATOIRE

```
0 CLS
10 CLEAR:DIMF(1700),G(1700):PRINT:INPUT"E.INF.,S.SUP.";E1,E5
20 MIN=E5:INPUT"NOMBRE DE TIRAGES ";H:CLS:FORR=1TOH
30 A=E3-RND(E5-E1)+1
40 A=INT(A):F(A)=F(A)+1:IFMIN>ATHENMIN=A
50 PRINTUSING" ######";A;:IFMAX<ATHENMAX=A
60 NEXTR:IFSQR(H)<15THEND=INT(SQR(H)+1)ELSED=15
70 MOY=MAX-MIN:C=INT(MOY/D+1):M=1
80 CLS:PRINT@7,"BORNE INF.: ";E1;"  -=0=-   BORNE SUP.: ";E5:PRINT
90 FORJ=INT(MIN)TOINT(MAX)STEPC:E=J+C
100 PRINTUSING"###";J;:PRINT" A ";USING"### ";E;
110 FORI=JTOJ+C-1:G(J)=G(J)+F(I):NEXTI
120 IFG(J)/H>47THENM=M+1:FORI=JTOMINSTEP-C:G(I)=0:NEXTI:GOTO80
130 IFG(J)>0THENFORL=1TOG(J)/M:PRINTCHR$(143);:NEXTL
140 IFG(J)=0THENPRINT:NEXTJ:GOTO160
150 PRINTINT(G(J)/H*100+.5);"%":NEXTJ
160 IFPEEK(14400)<>1THEN160ELSE10
```

- Modèle GAUSSIEN

```
0 CLS
10 CLEAR:DIMF(1700),G(1700):PRINT:INPUT"MOYENNE , EC.TYPE ";X,S
20 MIN=X:INPUT"NOMBRE DE TIRAGES ";H:CLS:FORR=1TOH
30 A=S*SQR(-2*LOG(RND(0)))*COS(6.28*RND(0))+X
40 A=INT(A):F(A)=F(A)+1:IFMIN>ATHENMIN=A
50 PRINTUSING" ######";A;:IFMAX<ATHENMAX=A
60 NEXTR:IFSQR(H)<15THEND=INT(SQR(H)+1)ELSED=15
70 MOY=MAX-MIN:C=INT(MOY/D+1):M=1
80 CLS:PRINT@7,"MOYENNE : ";X;"  -=0=-   ECART TYPE : ";S:PRINT
90 FORJ=INT(MIN)TOINT(MAX)STEPC:E=J+C
100 PRINTUSING"###";J;:PRINT" A ";USING"### ";E;
110 FORI=JTOJ+C-1:G(J)=G(J)+F(I):NEXTI
120 IFG(J)/H.49THENM=M+1:FORI=JTOMINSTEP-C:G(I)=0:NEXTI:GOTO80
130 IFG(J)>0THENFORL=1TOG(J)/M:PRINTCHR$(143);:NEXTL
140 IFG(J)=0THENPRINT:NEXTJ:GOTO160
150 PRINTINT(G(J)/H*100+.5);"%":NEXTJ
160 IFPEEK(14400)<>1THEN160ELSE10
```

- Modèle POISSONNIEN

```
0 CLS
10 CLEAR:DIMF(1600),G(1600):PRINT:INPUT"MOYENNE , EC.TYPE ";X,S
15 INPUT"CENTRAGE DE LA MOYENNE : A GAUCHE -3 A 0 , A DROITE 0 A +3 ";DE
20 MIN=X:INPUT"NOMBRE DE TIRAGES ";H:CLS:FORR=1TOH
30 A=S*SQR(-2*LOG(RND(0)))*(COS(6.28*RND(0))-CE/3)+X
40 A=INT(A):F(A)=F(A)+1:IFMIN>ATHENMIN=A
50 PRINTUSING" ######";A;:IFMAX<ATHENMAX=A
60 NEXTR:IFSQR(H)<15THEND=INT(SQR(H)+1)ELSED=15
70 MOY=MAX-MIN:C=INT(MOY/D+1):M=1
80 CLS:PRINT"MOYENNE : ";X;"  -=-   ECART TYPE : ";S;"  -=- CENTRAGE : ";CE;:PR
INT
90 FORJ=INT(MIN)TOINT(MAX)STEPC:E=J+C
100 PRINTUSING"###";J;:PRINT" A ";USING"### ";E;
110 FORI=JTOJ+C-1:G(J)=G(J)+F(I):NEXTI
120 IFG(J)/M*49THENM=M+1:FORI=JTOMINSTEP-C:G(I)=0:NEXTI:GOTO80
130 IFG(J)>0THENFORL=1TOG(J)/M:PRINTCHR$(143);:NEXTL
140 IFG(J)=0THENPRINT:NEXTJ:GOTO160
150 PRINTINT(G(J)/H*100+.5);"%":NEXTJ
160 IFPEEK(14400)<>1THEN160ELSERANDOM:GOTO10
```

PROGRAMME DE SIMULATION DES FILES D'ATTENTE

```
=======

0 '        ORIGINE L'O.I. N,I
5 CLEAR:DEFINT"DIM TX(100 ,T(100),FILE(1000 W=1:CLS
10 PRINT"* * *   E T U D E   D E S   F I L E S   D' A T T E N T E   * * *
  *      P R O G R A M M E   D E   S I M U L A T I O N      *"
12 PRINT:PRINT"                                    ---------------------"
15 PRINT:PRINT"    VISUALISATION DE L'HISTOGRAMME :

1- DES ATTENTES DE TRAITEMENT ( REPARTITION DE LA FILE )
2- DES ATTENTES D'ARRIVEE     ( REPARTITION DES ARRETS )"
20 PRINT:PRINT"   LEQUEL VOULEZ-VOUS OBSERVER ""
25 H$=INKEY$:H=VAL(H$)
30 IF H< 2 OR H> 1 THEN 25 ELSE CLS
40 PRINT"LES ARRIVEES OU LES TRAITEMENTS PEUVENT ETRE "

        1- MULTIPLES ( AVEC MODELE DE VARIATION A VENIR )
        2- SIMPLES   ( INDIVIDU PAR INDIVIDU )

40 EN EST-IL DANS VOTRE CAS ?... ";
45 MS$=INKEY$:MS=VAL(MS$)
46 IF MS< 1 OR MS> 1 THEN 45
47 IF MS=1 THEN PRINT"ARRIV  ET(OU) TRAIT  MULTIPLES"
48 IF MS=2 THEN PRINT"ARRIV  ET TRAIT, SIMPLES"
49 PRINT:PRINT"                                    ---=-="
50 PRINT:PRINT"IL VOUS RESTE MAINTENANT A DEFINIR :

        - LES INTERVALLES D'ARRIVEES      (   100 SI POSSIBLE )
        - LES INTERVALLES DE TRAITEMENT   (          "        )
        - L'EFFECTIF DE LA POPULATION     (   1000 IMPERATIF !)"
55 PRINT"PRINT".... A VOUS DE CHOISIR LES UNITES DE CALCUL .     ( ENTER S.V.P.)";
57 IF PEEK(14400)  1 THEN 57 ELSE CLS
60 PRINT"INTERVALLE D'ARRIVEE ' ";:O=1:GOSUB 6100:GOSUB 6400:VAR$=VR$(1):ITARIV=A
67 IF ITARIV>327 THEN A=1 ELSE P=100
68 IF VR=1 THEN PEVENT(1)=RND(ITARIV)
69 IF VR 1 AND VR 5 THEN PEVENT(1)=RND(ITARIV*P)/P
70 IF MS=1 THEN GOSUB 9010:CLS
```



```
71 PRINT"TEMPS DE TRAITEMENT ' ";:O=2:GOSUB 6100:GOSUB 6400:VTR$=VR$(1):TTRAIT=A
78 IF MS=1 THEN GOSUB 9020:CLS:PRINT:PRINT
80 INPUT"EFFECTIF DE LA POPULATION";EFFPOP:CLS
100 PEVENT(1)=10000 RANDOM
110 PRINT@96,"POUR VISUALISER LE DETAIL DE LA SIMULATION =  ENTER"
120 PRINT"                                          SINON =  CLEAR";
130 '---------------- BOUCLE DES EVENEMENTS ------------------
133 IF FILE G1 THEN G1=FILE
135 M=M+FILE:MN=MN+1
140 IF PEVENT(1)  PEVENT(2) THEN K=1
                            ELSE K=2
141 IF N>FILE =EFFPOP THEN I=2
142 IF PEEK(14400)=1 THEN V=1
143 IF PEEK(14400)=2 THEN V=2:CLS
144 IF V=0 THEN PRINT@470,"*** ";N;" + ";FILE;" ***   ";:GOTO 170
147 IF V=0 THEN 144
170 DATE=PEVENT(K)
180 ON K GOSUB 1000,2000
190 IF EVENEMENT=0 AND FILE 0 THEN GOSUB 3000
192 IF FILE=0 THEN E=E+1
195 IF E= 1 AND V  2 THEN GOSUB 8000
197 IF V=1 THEN V=0
200 IF EFFPOP N OR FILE 0 THEN 130
206 IF PEEK(14400)  2 AND V=0 THEN 206
210 GOSUB 4000
220 END

1000 '---------------- ARRIVEES ------------------
1005 GOSUB 6200:PEVENT(1)=DATE+A:IF MS=1 THEN GOSUB 9030:N1=INT(A+.5)
1010 FILE=FILE+1 E=0
1020 FILE(N+FILE)=DATE
1030 N1=N1-1:IF N1>0 THEN 1010
1999 RETURN
2000 -------- LE DERNIER TRAITEMENT EN COURS EST TERMINE -----
2020 PEVENT(2)=DATE+10000
2030 INOC=DATE
2035 IF FILE=0 THEN E=E+1:GOTO 2050
2050 EVENEMENT=0
2999 RETURN
3000 '---------------- UN NOUVEAU TRAITEMENT COMMENCE ----------
3010 GOSUB 6245:PEVENT(2)=DATE+A
3020 IF MS=1 THEN GOSUB 9070:N2=INT(A+.5)
3026 IF MS=1 AND AV>TR THEN 3028
3027 IF MS=1 AND V  A OR V  2 AND FILE(N+1)=DATE THEN GOSUB8000.GOTO3030
3028 IF MS=1 AND V  2 AND FILE(N+1)=DATE THEN GOSUB 8000
3030 ATTENTE=ATTENTE+(DATE-FILE(N+1))
3050 FILE=FILE-1:EVENEMENT=1:IF MS=1 THEN N2=N2-1 ELSE N2=0
3060 N=N+1
3062 IF DATE-FILE(N) G2 THEN G2=DATE-FILE(N)
3066 IF DATE INOC G3 THEN G3=DATE-INOC
3070 GOSUB 5000:IF N2 0 AND FILE 0 THEN 3030
3090 TINOC=TINOC+(DATE-INOC)
3999 RETURN
4000 '---------------- IMPRESSION DES RESULTATS ----------------
4005 CLS:FOR I=1 TO 4:IF VR$(I)="POISSONNIEN" THEN VR$(I)="POISSON,"
4010 NEXT I:PRINT TAB(29);"VAL. MOYENNE";TAB(42);"- MODELE DE VARIATION"
4020 IF MS=1 THEN PRINT"INT. D'ARRIV. / NB D'INDIV. :";ITARIV ELSE PRINT"INTERVALLE D'
ARRIVEE     :";
4025 IF MS=1 THEN PRINT TAB(28);INT(ITARIV);"/";INT(AV);"=";INT(ITARIV/AV*10+.5)
/10;TAB(44);VR$(1);TAB(53);" ";VR$(3) ELSE PRINT TAB(33);ITARIV;TAB(49);VAR$
4030 IF MS=1 THEN PRINT"TPS DE TRAIT. / NB D'INDIV. :";A ELSE PRINT"TEMPS DE TRAI
TEMENT     :";
4035 IF MS=1 THEN PRINT TAB(28);INT(TTRAIT );"/";INT(TR);"=";INT(TTRAIT/TR*10+.5)
/10;TAB(44);VR$(2);TAB(53);" ";VR$(4) ELSE PRINT TAB(33);TTRAIT;TAB(49);VTR$
4040 PRINT"EFFECTIF DE LA POPULATION   :   ";N
4045 PRINT:PRINT TAB(32);"MAXI";:PRINT TAB(42);"MOYEN"
4050 PRINT"NB. D'INDIVIDUS EN ATTENTE  :";
     PRINT TAB(3);G1;:PRINT TAB(41);FIX(M/MN+.5)
4055 PRINT"TPS D'ATTENTE DE TRAITEMENT :";
     PRINT TAB(31);G2, A=ATTENTE N*6:GOSUB 6000:PRINT TAB(41);A
4060 PRINT"TPS D'ATTENTE D'UN INDIVIDU :";
     PRINT TAB(31);G3,A=TINOC/N:GOSUB 6000:PRINT TAB(41);A
4065 PRINT"TEMPS MOYEN PAR INDIVIDU            ";A=PEVENT(2)/N:GOSUB 6000:PRINT
A
4070 PRINT"TEMPS TOTAL                         ";PEVENT(2)
4072 PRINT"TAUX MOYEN D'UTILISATION    :       ";A=(PEVENT(2)-TINOC)/PEVENT(2)*
100:GOSUB 6000:PRINT A;"%":PRINT
4074 IF HI=1 THEN PRINT"POUR VISUALISER L'HISTOGRAMME DES ATTENTES D'ARRIVEE =
ENTER
                                          SINON =' CLEAR"
4075 IF HI=1 THEN PRINT"POUR VISUALISER L'HISTOGRAMME DES ATTENTES DE TRAIT  =>
ENTER
                                          SINON => CLEAR"

4076 IF PEEK(14400)=2 THEN 4200
4078 IF PEEK(14400)  1 THEN 4076 ELSE CLS
4080 D=INT(SQR(N)+1):IF D 15 THEN D=15
4090 C=INT MAX/D+1:U=1
4100 CLS:IF HI=1 THEN PRINT@7,"***   HISTOGRAMME DES ATTENTES DE TRAITEMENT   **
*"
4110 IF HI=2 THEN PRINT@9,"***   HISTOGRAMME DES ATTENTES D'ARRIVEE   ***"
4115 PRINT
4120 FOR J=0 TO MAX STEP C:EC=J+C
4125   FOR I=J TO J+C-1:T(J)=T(J)+TX(I) NEXT I
4130 IF T(J) 0 THEN  PRINT USING"####";J;W;:PRINT" A ";USING"#### ";EC;W;
4150   IF T(J)>U 45 THEN U=U+1:FOR I=J TO 0 STEP -C:T(I)=0:NEXTI:GOTO4100
4160   IF T(J) 0 THEN FOR L=1 TO T(J)/U:PRINT CHR$(143);:NEXT L
4170   IF T(J)=0 THEN NEXT J
4180   PRINT INT(T(J)/N*1000+.5)/10;"%":NEXT J
4190 IF PEEK(14400)  1 THEN 4190
4200 CLS:PRINT:PRINT:PRINT:PRINT-PRINT"SI VOUS EN VOULEZ ENCORE , JE PEUX :
```

– le nombre d'individus en attente,
– les temps d'attente de traitement,
– les temps d'attente d'arrivée , puis en valeur unitaire :
– le temps moyen par individu,
– le temps total pour traiter le population,
– le taux d'utilisetion des moyens de traitement.

Enfin, une troisième partie donne la répartition des attentes de traitement ou d'arrivées suivant votre choix du début . c'est le trecé de l'histogramme des attentes.

L'ordinateur trace l'histogramme des attentes

Cheque classe de valeurs y est indiquée par ses bornes inférieure et supérieure et l'effectif y est exprimé de deux façons .

– un trait dont l'importance correspond à l'effectif (le programme est ainsi feit que, quel que soit l'effectif, une classe ne fera jamais plus d'une ligne ; si tel est le cas, on efface tout et l'on recommence, ce qui produit parfois quelques effets spéciaux...) ,

– à la suite du trait, le pourcentage que représente la classe per rapport à la population (si leur somme est différente de 100 %, ne vous posez pas trop de questions, c'est à cause des arrondis)

Après cela, il est possible (en enfonçant ENTER) de .

– revisualiser le tableau précédent,

– refaire une simulation avec les mêmes critères,

```
        1- VISUALISER LES RESULTATS PRECEDENTS
        2- FAIRE UNE NOUVELLE SIMULATION AVEC LES MEMES CRITERES          3-  ''
        ''        ''        AVEC D'AUTRES CRITERES"
4205 PRINT:PRINT"SINON ENFONCEZ LE  <  4 >."
4210 CH$=INKEY$:CH=VAL(CH$)
4220 IF CH=4 OR CH=3 THEN 4210 ELSE CLS
4230 IF CH=1 THEN FOR I=0 TO MAX:T(I)=0:NEXT I:GOTO 4000
4240 IF CH=2 THEN FOR I=0 TO MAX:TX(I)=0:T(I)=0:NEXT I:FOR I=0 TO N:FILE(I)=0:NE
XT I:N=0:V=0:M=0:MN=0:FILE=0:ATTENTE=0:TINOC=0:GOSUB 6200:PEVENT(1)=AIEV=0:G=0:I
NOC=0:G1=0:G2=0:G3=0:P1=0:P2=0:P3=0:GOTO 100
4250 IF CH=3 THEN 5
4260 IF CH=4 THEN PRINT@403,"VOUS ETES FATIGUES ' , , ,
     ALORS PRENEZ VOTRE TOUR DANS LA FILE D'ATTENTE  . .
                                  , , , CELA VOUS REPOSERA  "
4270 IF PEEK(14400)=1 THEN 4200 ELSE 4270
4999 RETURN
5000 '---------- COMPTABILISATION POUR HISTOGRAMME ------------
5010 IF HI=1 THEN X=INT(DATE+.5-FILE(N))
5020 IF HI=2 THEN X=INT(DATE+.5-INOC)
5030 X=X/W:IF X<0 THEN X=0
5040 IF X MAX THEN MAX=X
5050 IF MAX 100 THEN FOR I=1 TO 100:TX(INT(I/10))=TX(I)+TX(INT(I/10)):TX(I)=0:NE
XT I:W=W*10:MAX=MAX/10:GOTO 5000
5060 TX(X)=TX(X)+1
5999 RETURN
6000 '---------- ARRONDI A 2 DECIMALES , AU PLUS PRES ---------
6010 A=FIX((100*A)+.5)/100
6099 RETURN
6100 '---------- CHOIX DE LA VARIABILITE ----------------------
6110 PRINT"          1- FIXE
        2- ALEATOIRE
        3- GAUSSIEN
        4- POISSONNIEN
FAITES VOTRE CHOIX : ..."
6120 VP$=INKEY$:VP=VAL(VP$):IF VP<1 OR VP>4 THEN 6120
6130 IF VP=1 THEN VR$(0)="FIXE":PRINTVR$(0):INPUT"VALEUR ";A:RETURN
6140 IF VP=2 THEN VR$(0)="ALEATOIRE":PRINTVR$(0):INPUT"BORNE INF. , BORNE SUP. "
;B(0),B(1):GOTO6170
6150 IF VP=3 THEN VR$(0)="GAUSSIEN":PRINTVR$(0):INPUT"MOYENNE , ECART TYPE ";MO(
0),ET(0):GOTO 6190
6160 IF VP=4 THEN VR$(0)="POISSONNIEN":PRINTVR$(0):INPUT"MOYENNE , ECART TYPE ";
MO(0),ET(0):INPUT"CENTRAGE DE LA MOYENNE : A GAUCHE -3 A 0 , A DROITE 0 A +3 ";D
E(0):GOTO 6190
6170 IF B1(0)<B0(0) THEN B=B1(0):B1(0)=B0(0):B0(0)=B
6180 IF MO(0)<ET(0) THEN ME=MO(0):MO(0)=ET(0):ET(0)=ME
6190 RETURN
```

```
6200 '---------- TRI DES DONNEES VERS MODELE DE VARIATION -----
6210 IF VAR$="FIXE" THEN A=ITARIV:RETURN
6220 IF VAR$="ALEATOIRE" THEN D=1:GOTO 6320
6230 IF VAR$="GAUSSIEN" THEN D=1:GOTO 6330
6240 IF VAR$="POISSONNIEN" THEN D=1:GOTO 6340
6245 IF VTR$="FIXE" THEN A=TTRAIT:RETURN
6250 IF VTR$="ALEATOIRE" THEN D=2:GOTO 6320
6260 IF VTR$="GAUSSIEN" THEN D=2:GOTO 6330
6270 IF VTR$="POISSONNIEN" THEN D=2:GOTO 6340
6300 '------------------- VARIABILITES ----------------------
6310 A=(B1(0)*100+1-RND(0)*B1(0)*B0(0)*100+1)/100:RETURN
6320 A=ET(0)*SQR(-2*LOG(RND(0)))*COS(6.283*RND(0))+MO(0):GOTO 6000
6330 A=ET(0)*SQR(-2*LOG(RND(0)))*(COS(6.283*RND(0))-DE(0)/3)+MO(0):GOTO 6000
6400 '---------- AFFECTATION DE LA VARIABLE MOYENNE ------------
6410 IF VR=2 THEN A=(B1(0)+B0(0))/2
6420 IF VR=3 THEN A=MO(0)
6430 IF VR=4 THEN A=MO(0)
6999 RETURN
7000 '---------- INPRESSION DES DATES D'ARRIVEE EN ATTENTE ----
7005 H=0:H1=60
7010   FOR J=H TO H+H1 STEP 5
7020          FOR I=1 TO 5
7025            IF FILE(I+J) THEN PRINT:GOTO 7999
7030 IF FILE(N+I+J)>999 THEN PRINT USING"####.##",FILE(N+I+J), ELSE PRINT USING"
########";FILE(N+I+J);
7040          NEXT I
7045 IF FILE(I+J) THEN 7999
7050   PRINT:PRINT TAB(23);
7060   NEXT J:IF FILE 65 THEN 7999
7070 IF PEEK(14400)=1 THEN 7070 ELSE H=H+60:GOTO 7010
7999 L=INT((J+I-1)/5+.8):RETURN
8000 '---------- IMPRESSION DU TABLEAU DE SIMULATION ----------
8010 IF PEVENT(2) 999 THEN PRINT@128+9,USING"######.##";DATE:PEVENT(2);:PRINT"
";USING"###";(FILE:ELSE PRINT@128+9,USING"########";DATE;PEVENT(2);:PRINT"  "USI
NG"###";FILE;
8020 IF FILE  0 THEN GOSUB 7000 ELSE PRINT
8030 G=G+64*L:IF G+128 960 AND L=6 THEN PRINT
8040 IF G+128 960 THEN G=832
8050 PRINT@0,"        CALENDRIER          SITUATION DE LA FILE D'ATTENTE          ";P
RINT@64,"   DATES  FIN TRAIT.  NB          DATES D'ARRIVEE                  ";
8999 RETURN
9000 '------------ ARRIVEES ET TRAITEMENTS MULTIPLES ---------
9010 PRINT"NUMERE D'INDIVIDUS A VENIR SIMULTANEMENT : ";D=3:GOSUB 6100:GOSUB 640
0:AV=A:AV$=VR$(0):RETURN
9020 PRINT"NUMERE D'INDIVIDUS TRAITEES SIMULTANEMENT : ";D=4:GOSUB 6100:GOSUB 64
00:TR=A:TR$=VR$(0):RETURN
9030 IF AV$="FIXE" THEN D=3:A=AV:RETURN
9040 IF AV$="ALEATOIRE" THEN D=3:GOTO 6320
9050 IF AV$="GAUSSIEN" THEN D=3:GOTO 6330
9060 IF AV$="POISSONNIEN" THEN D=3:GOTO 6340
9070 IF TR$="FIXE" THEN D=4:A=TR:RETURN
9080 IF TR$="ALEATOIRE" THEN D=4:GOTO 6320
9090 IF TR$="GAUSSIEN" THEN D=4:GOTO 6330
9100 IF TR$="POISSONNIEN" THEN D=4:GOTO 6340
```

– refaire une simulation avec les critères différents.

Et si, de tout cele, rien ne répond à votre attente, enfoncez toujours le 4 ; sait-on jamais ?...

Exemple d'utilisetion

Question : combien faut-il mettre de postes de péage en fonction sur l'autoroute A 6 un retour de grand week-end ?

Données . on seit qu'une voie peut débiter une moyenne maximele de 2 000 véhicules par heure (soit 1,8 seconde par véhicule). (Cette valeur n'est qu'un exemple ; elle peut être plus élevée.)

Cette fréquence (1,8 seconde/véhicule) peut varier suivant le loi de Poisson (écart type 1 s, centrage de la moyenne – 2).

Le nombre d'errivées simultenées est lié au nombre de files, c'est-à-dire varie de 0 à 3 (moyenne 1,5, écart type 0,5).

Sachant qu'une caisse peut absorber 180 véhicules/heure (chiffre pouvant varier de 120 à 240 véhicules/h de façon aléatoire, soit 30 à 15 s/véhicule) et que nous avons une moyenne maximale de 2 000 véhicules/h arrivant en moyenne sur 1,5 file (soit 3 000 véhicules/h), nous pouvons donc en déduire le nombre de caisses à mettre en œuvre pour satisfaire le débit moyen d'arrivée, soit 3 000/180 = 17 caisses.

Il ne reste plus qu'à observer l'influence d'une telle décision sur le trefic...

Entrée des donnees

– visualisation de l histogramme → attente de traitement
– arrivees ou tratements → multiples
– intervalles d arrivée → Poissonnien ;
  • moyenne, écart type 1 8 1
  • centrage de la moyenne - 2
– nombre d arrivees simultanees → Gaussien (moyenne, écart type 1 5 0 5)
– temps de traitement → aleatoire (borne inferieure, borne supérieure 15, 30)
– nombre de traitements simultanes → fixe (valeur 17)
– nombre de voitures à simuler 1 000

Résultet	maximal	moyen
– nombre de voitures en attente	24	7
– temps d'attente des voitures	36 s	12 s
– temps d'attente de l ensemble des casses	1 s	0
– temps moyen por voiture	1 8 s soit 2 000 véhicules/h	
– temps total pour 1 000 voitures	1 760 s soit 30 min	
– taux d utilisation de l ensemble des casses	100 %	

Bien qu'ayant une capacité de traitements en moyenne inférieure aux arrivées (traitements 22,5/17 = 1,3 s/voiture, contre 1,8/1,5 = 1,2 s/voiture), nous observons une file d'attente négligeable (26 voitures pour 17 caisses) d'où la réalisation d'autres essais (tableau 1).

Le temps d'attente des caisses et le taux d'utilisation ne sont pas pris en compte, car le programme considère l'ensemble des caisses comme un tout et le fait qu'une seule soit utilisée vaut pour l'ensemble, ce qui, dans cet exemple d'utilisation, n'est pas représentatif (les cas d'une étude de stock ou d'utilisation machine, par exemple, correspondent mieux à ce programme).

On peut constater qu'au-delà de 17 caisses, le nombre de voitures et les temps sont stables (pas de différences significatives). Par contre, si l'on diminue le nombre de caisses, on observe une augmentetion des files d'attente immédiate (en nombre et en temps) sauf pour le débit (temps pour 1 000 voitures) où l'augmentation n'apperaît significative qu'à partir de 11 caisses.

Tableau 1

Nombre de caisses		40	35	30	25	20	17	15	13	12	11	10
Nombre de voitures en attente	max mal	21	23	23	23	23	24	31	44	132	204	222
	moyen	7	7	7	7	7	7	9	23	47	108	113
Temps d'attente des voitures	max mal	28	30	30	30	30	36	45	90	230	420	505
	moyen	11	12	12	12	12	12	16	39	84	224	257
Temps total pour 1 000 voitures (en secondes)		1 770	1 810	1 790	1 760	1 700	1 760	1 740	1 730	1 850	2 090	2 290

Tableau 2

Nombre de voitures en attente	maximal	67	95	111	60	85	146	103	67
	moyen	43	54	61	28	35	73	43	35
Temps d'attente des voitures	maximal	119	170	228	108	148	275	195	121
	moyen	78	100	116	50	64	141	80	65
Temps total pour 1 000 voitures (en secondes)		1 860	1 920	1 950	1 850	1 880	1 970	1 910	1 900

Nous pouvons envisager l'hypothèse de douze caisses plus en détail et voir les différents résultats obtenus (tableau 2).

Aucun bouchon à signaler (dans le pire des cas, treize voitures par caisse). Oui, me direz-vous, mais la simulation porte sur 1 000 voitures ; qu'en serait-il pour 10 000, 100 000 ? En d'autres termes, la file d'attente était-elle stabilisée ?

Pour le vérifier, deux possibilités sont offertes .

– regerder l'état de la file pendant la simuletion et observer que les plus fortes valeurs ne sont pes en fin de simulation ;

– ou, moins contraignant, visualiser « l'histogramme des attentes de traitements » et vérifier que les plus forts pourcentages ne coïncident pas evec les plus longues attentes.

Dans les simulations précédentes, seule la sixième peut laisser des doutes à ce sujet.

Conclusion : douze caisses peuvent être considérées comme le seuil à partir duquel le moindre augmentation de trafic crée un bouchon.

Gérard Nave

Tableaux des principales variables

I ARRIVEES

ITARIV	intervalle d arrivee
VAR$	type de variation de l intervalle d arrivee
AV$	type de variation du nombre d arrivées simultanées
AV	nombre d arrivées simultanées moyen

II TRAITEMENTS

TTRAIT	temps de traitement
VTR$	type de variation du temps de traitement
TR$	type de variation du nombre de traitements simultanes
TR	nombre de traitements simultanes moyens

III VARIABILITE

MS	choix des arrivees (ou traitements) simples ou multiples
VR$	type de variation
VR	numero de variation
O	affectation des calculs de variation
B1	borne inferieure
B0	borne supérieure
MO	moyenne
ET	ecart type
DE	decentrage de la moyenne

IV EFFECTIF

EFFPOP	effectif de la population
N	nombre d individus traites
N1	nombre d arrivees simultanées
N2	nombre de traitements simultanés
FILE	nombre d individus en attente
M	total des individus ayant attendu
MN	nombre d événements

V. DATE

DATE	date du dernier evenement
PEVENT (1)	date du prochain evenement → arrivee
PEVENT (2)	date du prochain evenement → fin de traitement
FILE (N)	date d arrivée des individus en attente
INOC	date du debut d inoccupation

VI TEMPS

ATTENTE	temps d attente total
TINOC	temps d inoccupation total

VII HISTOGRAMME

HI	choix de l histogramme (arrivée ou traitement)
TX (X)	nombre d attentes ayant la même valeur
X	valeur de l attente → adresse
T (X)	nombre d attentes par classe
MAX	valeur maximale d attente
W	diviseur pour reclassement des adresses
D	nombre de classes
I	intervalle de classes
EC	limite superieure de classe

votre argent vous intéresse, intéressez-vous aux intérêts

Que pensez-vous d'une société de ventes par correspondance qui vous propose, pour un objet de 1 490 FF, les facilités de paiement suivantes

3B4 FF comptant
12 mensualités de 112 FF

Cette société pratique, en fait, un taux d'intérêt annuel de 37,6 % ! ! et que pensez-vous de cet organisme de crédit qui vous accorde un prêt de 1 000 FF payable en 12 mensualités de 95,82 FF sans annoncer sur son contrat le taux d'intérêts qu'il pratique et qui est en réalité de 26,6 %

L'automobile que vous voulez acheter coûte 30 000 FF
La société X vous propose un crédit en 36 versements de 1 115 FF
La société Y, elle, propose un paiement comptant de 7 020 FF et 36 versements de 920 FF

Dans les deux cas, le total des intérêts versés est le même (10 140 FF). Pourtant le taux d'intérêt de la société X est de 20 % et celui de la société Y de 25,5 % Intéressant, n'est-ce-pas ?

Banques, organismes de crédit ? vous pouvez maintenant décider « Basiquement ».

Le programme ci-contre vous permettra de calculer avec précision le taux d'interêts composes annuel d'un emprunt, connaissant

la somme empruntée
le nombre de mensualités
la valeur de chaque mensualité

Le taux d'intérêt composé n'est, en principe, pas calculable directement Dans le programme ci-contre, il est fait appel à la méthode d'approximation de Newton, utilisée pour la détermination approchée des racines d'une équation de nature et de degré quelconques

La méthode appliquée est la suivante

recherche d'une valeur grossièrement approchée de la racine ;
affinage de la valeur de la racine ,
calcul d'une nouvelle approximation, et ainsi de suite Les approximations successives se resserrent jusqu'à obtention de la précision voulue (écart entre 2 valeurs successives)

Cette méthode , bien que n'étant pas universellement applicable (l'approximation initiale doit être suffisamment près de la racine), s'avère en pratique satisfaisante pour les problèmes de calcul de taux d'intérêt

Le PRINT USING du programme peut évidemment être remplacé par un PRINT ordinaire, dans le cas ou ce premier n'existe pas dans votre BASIC.

Alain Girpin

```
RUN
CALCULS DU TAUX D INTERETS COMPOSES
— NOMBRE DE MENSUALITES ? 12
— VALEUR DE LA MENSUALITE ? 112
— VALEUR DE L EMPRUNT ? 1106
— > TAUX D INTERET ANNUEL = 37 60%
AUTRE CALCUL ? OUI

— NOMBRE DE MENSUALITES ? 36
— VALEUR DE LA MENSUALITE ? 1115
— VALEUR DE L EMPRUNT ? 30000
— > TAUX D INTERET ANNUEL = 20 01 %
AUTRE CALCUL ? NON
 READY
10 REM PROGRAMME DE CALCUL DES INTERETS COMPOSES
20 REM AUTEUR  ALAIN GIRPIN
30 REM COPYRIGHT L ORDINATEUR INDIVIDUEL ET L AUTEUR
40 CLS : PRINT '--- CALCULS DU TAUX D INTERETS COMPOSES ---"
50 REM
60 PRINT : GOSUB 150 : REM ENTREE DES DONNEES
70 GOSUB 220 : REM CALCUL APPROXIMATION
80 GOSUB 280 : REM RECHERCHE VALEUR EXACTE
90 GOSUB 370 : REM EDITION DU TAUX D'INTERETS
100 REM
110 REM DEMANDE SI AUTRE CALCUL
120 PRINT " AUTRE CALCUL ", : GOSUB 9000
130 IF R$ = ' O ' THEN 60 : REM AUTRE PROBLEME
140 END : REM FIN DU PROGRAMME
150 REM
160 REM ENTREE DES DONNEES
170 REM
180 INPUT " -NOMBRE DE MENSUALITES ", N
190 INPUT " VALEUR DE LA MENSUALITE ", V
200 INPUT " -VALEUR DE L'EMPRUNT ", E
210 RETURN
220 REM
230 REM CALCUL DE L'APPROXIMATION
240 REM
250 R3 = E / V
260 R2 = 1 / R3 - (R3 / N ↑ 2)
270 RETURN
280 REM
290 REM RECHERCHE VALEUR EXACTE
300 REM
310 R5 = (R2 + 1) ↑ -N
320 X = R3 * R2 - 1 + R5
330 Y = (R5 * (N / (1 / R2 + 1) + 1) - 1) / R2
340 R2 = R2 + X / Y
350 IF ABS (X) >1 E - 6 THEN 310
360 RETURN
370 REM
380 REM EDITION DU TAUX D INTERET
390 REM
400 PRINTUSING " - >TAUX D'INTERET ANNUEL =###,##%  , R2 * 1200
410 RETURN
9000 INPUT R$ : REM OUI OU NON ?
9010 R$ = LEFT$ (R$ 1)
9020 IF R$<>' O ' AND R$<>' N ' THEN PRINT " O OU N ", : GO TO 9000
9030 RETURN
```


quand l'ordinateur gère votre trésorerie

Tout a commencé par la lecture des petites annonces d'appartements. Ceux qui me plaisaient semblaient vraiment hors de mes moyens financiers ; et ceux que je pensais pouvoir acheter étaient nettement trop petits. La seule solution était d'acheter un appartement de taille convenable, mais en mauvais état, et de répartir petit à petit les travaux au cours du temps. Mais ceci revenait à poser périodiquement la question : combien puis-je emprunter, et donc rembourser chaque mois, sans en être réduit à ne nourrir ma petite famille que de riz et de lentilles ? J'ai ainsi entrepris de faire l'étude détaillée des dépenses familiales et des recettes ; pour ces dernières, étant salarié, pas trop de problèmes, ni d'imprévu : salaire mensuel auquel il faut ajouter les allocations familiales plus des « primes » plus ou moins périodiques.

Avant de planifier des dépenses « extraordinaires » il faut commencer par bien connaître les dépenses « obligatoires » qui sont variées tant en montant qu'en périodicité

Une étude du passé récent est très instructive et permet de chiffrer et de dater les divers éléments.

L'habitation : un loyer (ou un remboursement d'emprunt) plus les charges et les impôts associés (il y en a 4 ou 5 pour un propriétaire)

L'alimentation : les achats, dans les diverses boutiques, sont assez difficiles à apprécier exactement, surtout a posteriori lorsqu'ils ont été réglés en argent liquide Mais, devant l'importance du montant relatif, il faut examiner très soigneusement le montant de ce poste (ne pas oublier les prix des repas de cantines d'entreprise ou d'école)

Les services : téléphone, EDF-GDF, éventuellement eau , en tout, au moins une facture par mois, plus la femme de ménage

Les vacances : trajets, logements sur place, nourriture pour les divers éléments de la famille (les enfants et les parents ne suivent pas toujours le même itinéraire)

Les vêtements : pour chacun des membres de la famille Les gros achats réglés par chèques sont faciles à analyser Les achats courants réglés en liquide, ou les achats dans les grandes surfaces risquent de se retrouver englobés dans le budget alimentation

L'auto (ou les autos) achat (ou remboursement des mensualités), garage, entretien (essence, graissage, pneus).

Les enfants : montant des pensions, de la rentrée des classes (un pic !)

Les sports : clubs sportifs, matériel et trajets (chaussures, vêtements, skis, bateaux)

Les objets de loisir : livres, revues, journaux, musique, bricolage, cadeaux (Noël et anniversaires)

Et ne pas oublier **les impôts** ils sont assez faciles à repérer, plus difficiles à prédire, surtout au-delà de 12 mois (les montants croissent, les dates se rapprochent du début de l année)

Quand on fait le cumul de toutes les dépenses, on voit apparaître certains pics à des dates fixes, ce qui confirme que la simple gestion

mensuelle amène inexorablement des problèmes de trésorerie. Il faut néanmoins s'arranger pour avoir toujours un solde créditeur et, si l'on est très prudent, supérieur à une certaine somme en cas d'ennuis imprévus accident ou maladie grave.

Des prévisions en général difficiles

Ainsi, par un dépouillement soigneux des diverses opérations enregistrées sur les comptes postaux ou de chèques, il est possible de connaître les dates et les montants des divers éléments, tant en recette qu'en dépense. On peut donc, sans trop de difficulté, faire la liste et l'échéancier de toutes les dépenses que l'on est plus ou moins obligé d'effectuer, ainsi que des ressources dont on va disposer · avec ou sans calculette (mais sa présence diminue le nombre d'erreurs !) on peut obtenir une assez bonne prévision — tel jour il doit rentrer telle somme, tel jour tant — et, ainsi, connaître le montant disponible en fonction du temps.

En opérant ainsi, on s'aperçoit vite qu'à côté des dépenses clairement répertoriées, il y a une foule de petites dépenses qui se font à longueur d'année et n'apparaissent dans aucun compte car elles sont individuellement faibles : tickets de métro, courrier, journaux, nettoyage des vêtements, cinéma, musées, expositions, taxis, quêtes, pourboires divers.

Toutes sommes qui correspondent généralement à des dépenses en argent liquide, dont il faut exclure les dépenses d'alimentation, sinon on arrive à une somme importante difficile à maîtriser.

Pour toutes ces dépenses, un montant forfaitaire journalier s'avère une bonne solution.

Ceci se fait en ajoutant à toutes les dépenses énumérées ci-dessus une dépense « constante », un peu analogue à une petite fuite dans le montant des disponibilités, qui permet de serrer la réalité de façon réaliste, mais ne simplifie pas les calculs (combien y-a-t-il de jours du 23 mars au 7 avril ?).

En opérant ainsi, on obtient des résultats assez bons, d'autant meilleurs que les hypothèses sont plus précises.

Connaissant les ressources et dépenses « normales », on peut alors aborder l'étude des dépenses « extraordinaires » — ou d'une variation des ressources — Dans le cas cité plus haut, il s'agit essentiellement des travaux d'aménagement du logement une fois que l'on y est installé.

C'est alors que les calculs deviennent vraiment fastidieux, car il est nécessaire de faire plusieurs hypothèses avant d'arriver à une solution acceptable. Par ailleurs, les « conditions environnantes » évoluent : on renonce pour cette année à refaire la cuisine, ce que l'on avait prévu en juin (car le réfrigérateur et la machine à laver sont à bout de souffle), et l'on va profiter des réductions du salon des Arts ménagers (en mars).

Mais va-t-on régler comptant ou bien en 6 ou 12 mensualités ? Il faut alors modifier tous les calculs en ne prenant pas en compte les montants correspondant à la cuisine, mais ceux correspondant aux appareils ménagers, avec deux ou trois séries d'hypothèses.

Et là, on s'aperçoit qu'il est vraiment pénible de gérer au mieux un budget toujours à la limite de l'équilibre.

Laissez la machine faire les calculs

C'est à cause de cette deuxième phase que j'ai mis la prévision budgétaire sur mon ordinateur individuel, en adaptant la méthode qui avait fait ses preuves à la main, mais en faisant faire par la machine toute la partie fastidieuse du travail : les machines sont faites pour ça !

Les données qui sont immuables, ou tout au moins que l'on peut considérer comme telles car on n'y peut rien, sont stockées sur cassette une fois pour toutes (en les introduisant avec un grand soin, car il ne sera plus possible de les modifier) : ce sont les données détaillées ci-dessus.

L'idée de base est de pouvoir faire des simulations successives du budget prévisionnel en introduisant au clavier les hypothèses qui risquent de varier puis, lorsque l'on a choisi une politique, d'en enregistrer les éléments sur la cassette, et ainsi de suite.

Pour cela le programme débute par une demande de conditions initiales, date à partir de laquelle on veut simuler (inutile de chercher à simuler le passé) et le montant disponible à cette date (ce qui est une remise à jour très pratique des comptes).

L'ordinateur bouc-émissaire

Puis, si l'on dispose d'une cassette contenant des données déjà stockées, on lit cette cassette où l'on trouve, pour chaque opération, son nom, sa date, son montant et sa nature (recette ou dépense). Au cours de cette lecture on élimine les enregistrements correspondants à une date antérieure à la date minimum fournie au début. Les autres données sont stockées en mémoire.

On ajoute alors au clavier des opérations supplémentaires, que l'on peut ou non ajouter sur la cassette à la suite de celles déjà lues, ce qui permet d'étudier des hypothèses successives (que l'on ne stocke pas sur cassette) puis un choix (que l'on stocke).

On peut, de même, étendre le champs des prévisions en ajoutant les recettes et les dépenses pour des périodes de temps plus lointaines ; une application de cette méthode est la création de la première cassette : partant de rien, on ajoute les premières opérations.

Enfin, on introduit la valeur de la dépense « constante » définie plus haut par son montant mensuel, si l'on désire en introduire une.

Tous les éléments sont alors dans la mémoire de l'ordinateur, qui trie les enregistrements en fonction de leur date, puis effectue tous les calculs et affiche les résultats.

A la fin de ce cycle, l'utilisateur connaît clairement les conséquences prévisibles de ses hypothèses. Et la réalité confirme souvent les prévisions ainsi faites, dans la mesure où cette prévision n'est pas à trop long terme (au-delà de 18 à 20 mois il faudrait tenir compte de diverses variations de prix) et surtout dans la mesure où les données introduites sont sérieuses.

Comme il s'agit de la gestion prévisionnelle du budget d'un ménage, les notions telles que les amortissements n'ont pas leur place (si ce n'est sous la forme indirecte du montant des remboursements correspondant aux emprunts faits).

Des prévisions de six à douze mois

Le programme a été tout d'abord écrit sur un Tandy TRS 80 (4 K, Basic Level 1) puis il a été transformé pour un TRS-80 avec BASIC Level II. C'est ce dernier programme qui est décrit ici. Je vous fait grâce de la liste de la première version qui comportait de nombreuses acrobaties, nécessaires pour faire tourner le programme en Basic Level 1.

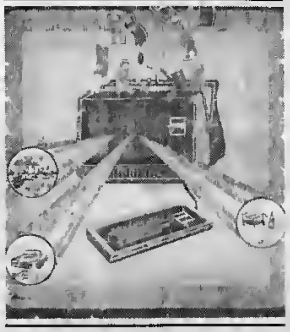
Parmi les contraintes, les deux plus sévères sont la taille mémoire et le fait de ne disposer que d'une unité de lecture-écriture sur cassette. La cassette n'ayant pas de retour arrière sur un enregistrement (le « BACKSPACE » de certains langages) il n'est pas possible d'avoir un enregistrement de fin de fichier.

On en est réduit à connaître la nombre d'enregistrements déjà écrits, à les lire, et à passer en mode enregistrement sans modifier la position de la cassette pour allonger le fichier, et enfin à recueillir soigneusement le nouveau nombre total d'enregistrements pour le passage suivant.

La taille mémoire limite le nombre d'opérations que l'on peut prendre en compte. Pour un TRS 80 4K Level 1, on arrive à une soixantaine d'enregistrements, soit 10 mois à la cadence de 6 par mois. En Level II, avec la même taille mémoire, le fait de conserver les noms des opérations conduit à ne pouvoir en traiter que 35 à 40, soit une prévision de 6 mois, ce qui est faible. (En passant à 16 K, il n'y a aucun problème pour traiter une prévision raisonnable).

Dans les deux programmes donnés ici, le manque de place mémoire a entraîné la suppression de tous les commentaires, ce qui rend le texte difficile à comprendre. De même, de nombreuses variables sont réutilisées, ce qui économise la place, mais diminue la clarté, et doit être évité de façon générale.

Trois tableaux C, D, et M contiennent, pour chaque opération, son



nom, sa date et son montant respectivement. Le nom est une chaîne limitée à 16 caractères, la date est condensée sous la forme d'un entier (positif et inférieur à 22 000), le montant est sous la forme d'un nombre en virgule flottante avec signe (+ signifiant recette, — dépense).

Les ordres commençant par des numéros inférieurs à 500 forment le programme principal, les suivants divers sous-programmes.

Six sous-programmes

De 600 à 630, il y a un sous-programme créant la date composée à partir du jour, du mois et de l'année. Cette date composée est analogue à celle obtenue en mettant bout à bout les deux derniers chiffres de l'année, du mois et du jour (exemple pour 15 septembre 1978, on obtient 780915). On obtient ainsi des nombres qui croissent dans l'ordre des dates, ce qui facilite le tri.

Pour pouvoir faire tenir ceci dans une variable entière (et gagner ainsi de l'espace mémoire) il ne faut pas dépasser 32 767. Pour cela l'année est remplacée par sa valeur moins 78. (Ainsi le 1er janvier 1978 devient 000101 et le 30 décembre 1999 : 211230). Pour gagner de la place en mémoire tous les mois ont été supposés égaux, et de 30 jours.

En 650 figure un sous-programme effectuant l'opération inverse, c'est-à-dire redonnant l'année, le mois et le jour à partir de la date composée.

En 680 figure un sous-programme fournissant le titre de l'édition finale de l'évolution de la trésorerie.

En 690, un sous-programme imprime une ligne d'étoiles (*). Le programme d'édition finale, sur lequel nous reviendrons plus loin, couvre les lignes 700 à 870.

En 950-960 on trouve le sous-programme imprimant une ligne du tableau d'évolution de trésorerie.

De 9 000 à 9 020 figure le sous-programme habituel de vérification d'une réponse OUI ou NON.

Revenons au programme principal

Jusqu'en 50 : obtention de la date minimum (E) et du montant initial (MI) ainsi que diverses initialisations.

de 60 à 120 : lecture du contenu d'une cassette provenant d'un traitement précédent, et cours de la lecture, on élimine les opérations antérieures à la date minimum (H < E), en ne les conservant pas en mémoire dans les tableaux des noms (C), des dates (D) et des montants (M). Les enregistrements retenus sont affichés au fur et à mesure de la lecture sur l'écran, ce qui fait patienter l'utilisateur et lui permet de contrôler le bon déroulement des opérations.

de 150 à 310 : acquisition au clavier et traitement des données relatives à des opérations supplémentaires, acquisition et contrôle entre 190 et 220, stockage en mémoire en 230, et (éventuellement) sur cassette en 240. Les ordres 280 à 310 donnent le nombre total d'enregistrements stockés sur la cassette, avec un cadre spectaculaire pour éviter qu'il ne passe inaperçu. (On peut gagner de la place en mémoire, *au détriment de la sécurité* en réduisant l'ampleur de ce cadre).

de 350 à 370 : il y a l'introduction de la « dépense constante ».

en 400 et **410** sont effectuées diverses initialisations pour préparer l'édition des résultats.

en 420 à 450 est effectué le tri des arguments dans l'ordre des dates croissantes : le tri détermine

```
10 DEFSTR A-C DEFINTD-L R-Z
20 DIM(12) C(50) D(50),M(50)
30 PRINT"PREVISION BUDGETAIRE" PRINT
40 PRINT"DATE MINIMUM A PRENDRE EN COMPTE" GOSUB600
50 INPUT"MONTANT DISPONIBLE A CETTE DATE",MI E=N J=0 L=0 PRINT
60 PRINT"AVEZ-VOUS UNE CASSETTE A LIRE"  GOSUB9000 Z$=R$ IFZ$="N"THEN150
70 PRINT"MONTEZ LA CASSETTE SUR LE MAGNETOPHONE ET POSITIONNEZ EN LECTURE"
80 INPUT"NOMBRE TOTAL D ENREGISTREMENTS DEJA ECRITS",L CLS
90 GOSUB680 FOR I=0TOL-1 INPUT#-1,C(I),D(I),M(I) H=D(I) IFNCETHEN120
100 GOSUB650 N=M(I) IFN>=0THENA="CR"ELSEH="D" N=-N
110 PRINTUSING" %                %  ##/##/##  ##########  ##",C(I),Y,X,W,N A J=J+1
120 NEXTI
150 PRINT"VOULEZ-VOUS AJOUTER DES DONNEES", GOSUB9000 IFR$="N"THEN350
160 PRINT"DESIREZ-VOUS LES CONSERVER SUR CASSETTE"  GOSUB9000 T$=R$
170 CLS PRINT IFZ$="N"ANDT$="O"THENPRINT"MONTEZ UNE CASSETTE VIERGE"
180 IFT$="O"THENPRINT"APPUYEZ SUR PLAY ET PECORD"
190 INPUT"NOM DE L OPERATION",A A=LEFT$(A 16) PRINT"SA DATE"  GOSUB680
200 IFHCETHENPRINT"DATE ANTERIEURE AU MINIMUM" GOTO190
210 INPUT"      MONTANT"  N N=ABS(N)
220 INPUT"CREDIT OU DEBIT",B IFB="CREDIT"THENB=1
ELSEIFB="DEBIT THENB=-1
ELSE220
230 N=N*B D(J)=N M(J)=N C(J)=H J=J+1
240 IFT$="O"THENPRINT# 1,A,H,N L=L+1
250 PRINT"AVEZ-VOUS D AUTRES DONNEES" GOSUB9000 IFR$="O"THEN190
260 IFT$="N"THEN350
280 GOSUB690
290 PRINT"PETENEZ SOIGNEUSEMENT LE NOMBRE TOTAL D ENREGISTREMENTS"
300 PRINT">>>>>>>>>>>>>>>> IL EST DE"L"<<<<<<<<<<<<<<<<"
310 GOSUB690
350 PRINT"VOULEZ-VOUS INTRODUIRE UNE DEPENSE CONTINUE"  GOSUB9000 U$=R$
360 IFU$="O"THENINPUT"SON MONTANT MENSUEL" N G=-ABS(N)/30
400 H=E GOSUB650 GOSUB690
410 PRINT#67   PRINTUSING"MONTANT INITIAL   ##/##/##                  ##########",Y,X,W,MI
420 I=J-1 V=0 FOR J=0TOI-1 H=D(J)
430 FORK=J+1TOI E=D(K)
440 IFH>EETHENDD(J)=E D(K)=H H=E M=M(J) M(J)=M(K) M(K)=M A=C(J) C(J)=C(K) C(K)=A
450 NEXTK E=W F=X G=Y GOSUB700 NEXTJ
460 A=C(I) H=D(I) N=M(I) E=W F=X G=Y GOSUB700
500 END
600 INPUT"JOUR(ENTRE 1 ET 30)",Y IFY<1ORY>30THEN600
610 INPUT"MOIS(ENTRE 1 ET 12)",X IFX<1ORX>12THEN610
620 INPUT"ANNEE(ENTRE 78 ET 99)",W IFW<78ORW>99THEN620
630 W=W-20 H=W*10000+X*100+Y RETURN
650 W=H/10000 Y=H-W*10000 X=Y/100 Y=Y-X*100 W=W+78 RETURN
680 CLS PRINT# "INITIALE", PRINT#24,"DATE"  PRINT#37 "MONTANT"  PRINT#46,"SOLDE"  RETURN
690 FORI=1TO61 PRINT#*' NEXTI9 PRINT RETURN
700 GOSUB650 IFU$="N"THEN720
710 N=(W-E)*360+(X-F)*30+Y-G MI=MI+N*Q
720 MI=MI+M(J) IF I>12THEN900
740 P(J)=MI L=I+1 S=J R=I GOSUB900 GOTO870
800 IFV<0THEN850
810 FORL=8TO7 P(L)=P(L+5) NEXTL PRINT"SUITE", GOSUB9100 GOSUB690
820 FORL=8TO7 S=J-8+L H=D(S) GOSUB650 K=L GOSUB950 NEXTL
830 H=D(S) GOSUB650
850 L=8+V R=L P(L)=MI S=J GOSUB950 V=V+1 IFV=5THENV=0
870 RETURN
900 PRINT#64+64+L,,
950 PRINTUSING %                %  ##/##/##  ##########  ##########",C(S),Y,X,N M(S) P(R) RETURN
9000 INPUTR$
9010 IFR$<>"O"ANDR$<>"N"THENPRINT"O OU N" GOTO9000
9020 RETURN
```

successivement la première opération à prendre en compte dans l'ensemble des opérations restantes. Aussitôt que cet argument est obtenu, il est transmis au sous-programme d'édition et affiché sur l'écran ce qui présente deux avantages : pour l'utilisateur, il est désagréable de rester devant un écran vide où il ne se passe rien : cette méthode morcelle le temps d'attente ; pour l'économie de place mémoire : le tableau P qui contient les soldes successifs peut être réduit à la dimension d'un écran (soit 13 éléments).

Le sous-programme d'édition est formé de deux parties : la première pour éditer le tableau initial, la seconde pour éditer les tableaux suivants. La technique d'édition par tableaux successifs permet à l'utilisateur d'avoir devant les yeux une image stable aussi longtemps qu'il le désire, plus exactement tant qu'il n'appuie pas sur la touche ENTER. Un recouvrement important se fait d'un tableau au suivant, ce qui offre l'avantage de pouvoir examiner de près ce qui se passe entre deux tableaux successifs. Les lignes 700 à 740 permettent la création du premier tableau : les données sont affichées aussi rapidement que la technique de tri les fournit, jusqu'à l'obtention d'un écran plein. Les lignes 800 à 870 exécutent la création des tableaux successifs suivants par reprise des huit lignes finales du tableau précédent, et affichage de 6 lignes suivantes au fur et à mesure de leur obtention par le tri.

Les critiques sur ces programmes sont de natures différentes suivant que l'on considère que l'on travaille sur une machine de 4 K de mémoire vive ou plus.

Des perfectionnements qui dépendent de la mémoire

Avec 4 K de mémoire on peut chercher à simplifier l'édition finale. On remplace alors les ordres 700 à 870 par l'édition d'une ligne (analogue aux ordres 950 et 960) avec un compteur pour arrêter l'édition toutes les 10 lignes par exemple (sinon le défilement trop rapide ne permet pas de consulter les résultats), la reprise se faisant en tapant sur la touche ENTER.

Ceci permet aussi de supprimer tout le tableau P. On peut ainsi gagner en nombre maximum d'opérations traitées, environ 13 opérations de plus.

On peut aussi diminuer la taille du cadre utilisé lors de l'affichage du nombre total d'enregistrements figurant sur la cassette : mais un cadre moins spectaculaire augmente les chances d'oubli de ce fameux nombre d'enregistrements stockés sur la cassette... **Il faut choisir entre la sécurité et la performance.**

Avec plus de mémoire on peut ajouter un certain nombre de perfectionnements.

Un générateur automatique des opérations à répétition périodique (cas d'un remboursement mensuel par exemple), ceci évitant d'avoir à refrapper la même chose pour tous les mois.

La possibilité, à la fin d'une exécution, de vider la totalité des opérations contenues en mémoire sur la cassette afin de simplifier les opérations après un choix de solution.

Visualisation des fluctuations sous forme d'un graphique.

Prise en compte de la longueur exacte des mois, tant dans la saisie des dates que dans le calcul des différences de dates.

Trouver et afficher la date où le montant du solde est le plus faible, et la valeur de ce montant.

Christophe Disabeau

lundi, mardi, mercredi, jeudi, vendredi... vous rappelez-vous le "jour" de votre naissance ?

20 lignes de basic pour vous rafraîchir la mémoire

L'article qui suit vous présente un petit programme en langage BASIC qui répondra à cette question, ainsi qu'à celles-ci par exemple :

— Quel était le jour de la prise de la Bastille ?
— Edmond Rostand dit dans Cyrano de Bergerac : « Et samedi vingt-six (septembre 1655), une heure avant le dîné, « Monsieur de Bergerac est mort assassiné. » Cet événement peut-il être authentique ? Mais d'abord, examinons quelques notions de base.

Nous vivons depuis l'année 1582 sous le règne du calendrier grégorien. Cette année-là, et afin de rattraper l'avance créée par le précédent système de calendrier (julien fondé par Jules — d'où le nom — César), le pape Grégoire XIII — dit Le Grand — décida que le lendemain du jeudi 4 octobre serait le vendredi (ouf !). 15 octobre ! De plus, afin de maintenir l'année moyenne à une valeur plus proche de l'année tropique (365,2425 jours contre 365,25 dans le système julien), il fut décidé que les années séculaires ne seraient plus bissextiles, sauf celles dont la valeur est multiple de 400, ce qui fait gagner 3 jours tous les 400 ans. Évidemment, la perfection n'étant pas de ce monde, le présent calendrier accusera encore trois jours de trop... mais d'ici 10 000 années. Le programme propose devra alors être modifié en conséquence.

Bien entendu, la mise en usage de la réforme grégorienne ne fut pas uniformément acceptée par tous les pays. Ainsi, en France, le retrait des 10 jours fautifs eut lieu entre le 9 et le 20 décembre 1582, aux Pays-Bas entre le 14 décembre et Noël, l'Allemagne et la Suisse (catholique) acceptèrent la réforme deux ans plus tard, la Pologne 4 ans, la Hongrie 5 ans et l'Angleterre 170 ans, soit en 1752 : mais il leur fallut sacrifier non pas dix, mais onze jours... ce qui donna lieu à de vives protestations ! Il est vrai, nous rappelle Paul Couderc, que le début de la même année avait été avancé de 3 mois !

Revenons à notre programme.

Celui-ci, à partir d'une date introduite en système julien ou grégorien, donne deux informations :

. *Le numéro du jour julien* : c'est une convention astronomique représentant le nombre de jours écoulés depuis le 1er janvier 4713 avant J.-C. (— 4712). Ce nombre sert de base au calcul du jour de la semaine et permet, en outre, de connaître le nombre de jours entre 2 dates en faisant la différence de leurs nombres juliens respectifs.

. *Le jour de la semaine* correspondant à la date donnée.

Il est à noter que les dates introduites avant la réforme grégorienne doivent être données en système julien et qu'elles peuvent être négatives et remonter jusqu'en — 4712 (4713 av. J.-C.). Dans l'autre sens, elles peuvent dépasser notre époque et atteindre des dates futuristes... jusqu'à la prochaine réforme...

Pour les amateurs, voici les formules :

JD = J + ENT (367 (MP - 1 + 5)/12) + ENT (365,25 (AP + 4712)) + Z

avec

J = jour, M = mois, A = année

MP = M + 12 si M = 1 ou 2
sinon MP = M

AP = A — 1 si M = 1 ou 2
sinon AP = A

Z = 2 pour dates juliennes,
= ENT [AP/400] — ENT [AP/400] pour dates grégoriennes
ENT [] signifie partie entière de X

. *Jour de la semaine* (0 à 6 avec 0 = dimanche, 1 = lundi, etc.)

JS = 7 × FRAC [(JD - 1720977)/7]

(FRAC signifie partie fractionnaire de [])

Et maintenant :

. Quel était le jour de la prise de la Bastille ?
— Le 14 juillet 1789 était un mardi.

. Edmond Rostand avait-il raison ?
— Non ! Le 26 septembre 1655 était un dimanche.

Alain Girpin

GLOSSAIRE

Année tropique : *temps séparant deux passages consécutifs du soleil au tropique. Elle vaut 365,24220 jours.*

Année séculaire : *année dont le millésime se termine par deux zéros (Exemple : 1800, 1900, 2000...).*

Année bissextile : *année dont le millésime est divisible par 4 (1972, 1976, 1980...).*

BIBLIOGRAPHIE

Le calendrier, Paul Couderc, collection « Que Sais-je ». Puri-sciences, 1979, Encyclopedia Universalis.

Première parution : L'OI n° 5

Cette fiche pratique donne le programme annoncé dans l'article précédent qui permet de déterminer, à partir de n'importe quelle date, le jour de la semaine correspondant.

Le jour Julien JD a une valeur de l'ordre de 365 × (1973 + 4713) = 2 442 215. Sa valeur s'exprime donc avec 7 chiffres, on dit que cette valeur a sept chiffres significatifs.

Un BASIC standard donne une précision insuffisante pour effectuer les calculs mis en jeu ici, puisque la précision des calculs en BASIC standard est de l'ordre de 6,5 chiffres de précision, alors qu'en fait, ici, il en faut beaucoup plus de 7.

Le programme donné ici est écrit avec le BASIC étendu de Microsoft, qui permet d'effectuer des calculs en double précision, c'est-à-dire avec 16 chiffres significatifs. Ceci se fait à condition que l'on fasse suivre le nom de la variable (ici, JD et JS) du signe # Sur le PET, la précision est de 12 chiffres significatifs pour tous les calculs.

Afin de limiter les erreurs de calcul dues aux arrondis, on rajoute 0,001 aux nombres dont on prend la partie entière (lignes 90, 130 et 140).

Le sous-programme 130-170 calcule JD à partir du jour J, du mois M et de l'année A, suivant les formules déjà données.

Après l'appel en 60 de ce sous-programme, on peut alors calculer en 80 et 90 le jour dans la semaine (0, 1, 2, ... 5 ou 6).

Il ne reste plus alors en 100 qu'à lire, par une instruction READ exécutée autant de fois que nécessaire, le nom du jour que l'on a mis en DATA à la ligne 180.

L'instruction RESTORE de la ligne 100 a pour effet que le premier X$ lu sera toujours DIMANCHE.

```
5 REM PROGRAMME DE CALENDRIER
7 REM AUTEUR ALAIN GIRPIN
10 REM COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
20 PRINT"<<<-CALENDRIER->>>": PRINT:PRINT
30 INPUT "SYSTEME JULIEN OU GREGORIEN", S$
35 IFS$="FIN" THEN STOP
40 S$=LEFT$ (S$,1): IF S$<>"J" ANDS$ <>"G" THEN 30
50 INPUT "DATE (JOUR, MOIS, ANNEE) ", J,M,A: PRINT
60 GOSUB 130
70 PRINT " -- NO DU JOUR JULIEN=", JD # ; ";",
80 JS # = (JD # -1720977)/7
90 JS # = INT (7*(JS # INT(JS # )) + 001)
100 RESTORE: FORN= 0TOJS # : READX$: NEXT N
110 PRINT "JOUR DE LA SEMAINE= ";X$: PRINT
120 GOTO30
130 MAN=INT (6+1/M + 001): MP=M +12*MAN:
    AP=A-MAN
140 JD # =J+INT ((367*(MP-1)+5)/12 + 001) + INT (365 26*
    (AP-1+4712)+ 001)
150 IFS$="J" THENJD # =JD # -2: GOTO 170
160 JD# =JD# INT (AP/100)+INT (AP/400)
170 RETURN
180 DATA DIMANCHE, LUNDI, MARDI, MERCREDI,
    JEUDI, VENDREDI, SAMEDI
```


rendez-vous avec votre ordinateur

grâce à ce programme tout simple

Chaque jour, je me lève en pensant à tout ce que je dois faire dans la journée... et chaque soir, je suis fâché avec moi-même. Pourquoi ? eh bien, j'ai oublié de téléphoner à telle personne... ou bien, je n'ai pas pu terminer tel travail... ou bien... Je ne sais pas si vous êtes comme moi, mais je suis souvent plus préoccupé par les tâches que je n'ai pas pu accomplir, que par celles qu'il me reste à réaliser. Alors, un beau jour, j'ai décidé de résoudre définitivement ce problème pour le moins épineux.

J'ai tout d'abord observé ce qui se passait. Je me définissais, pour chaque semaine, un certain nombre de tâches à effectuer, mais sans évaluer la durée de chacune d'elle. Souvent, il m'aurait en fait fallu, soit travailler 24 heures sans interruption, soit faire une dizaine de choses en même temps, afin d'honorer mes prévisions arbitraires. Un vieux dicton affirme qu'il ne faut pas courir deux lièvres à la fois, et je ne peux pas plus avoir deux activités en un même temps !

Je n'ai eu aucune difficulté à m'apercevoir que l'une des causes les plus importantes de mon problème était sans doute l'absence de plan de travail.

Une solution m'est alors apparue absolument évidente : je devais m'imposer une certaine discipline. Pour cela, le meilleur moyen était d'élaborer un planning clair qui soit une représentation assez fidèle des tâches quotidiennes à accomplir. Il ne faut toutefois pas « verser dans l'extrême » qui consiste à mettre en place une programmation détaillée de ses activités, qui serait alors paralysante.

Le résultat, c'est que maintenant je suis plus efficace pour la réalisation de l'ensemble de mes activités et ceci pour deux raisons principales.

Suivre une ligne de conduite évite souvent de s'écarter de l'objectif que l'on s'est fixé. En effet, j'avais une tendance naturelle à vouloir me consacrer à 36 affaires différentes sans en achever une seule.

Un problème bien posé est à moitié résolu. En effet, ce planning doit être avant tout réaliste, et ceci m'impose donc souvent de procéder à une analyse plus précise des éléments qui concourent à l'accomplissement d'un travail donné : est-il nécessaire, par exemple, d'accomplir telle tâche aujourd'hui plutôt que telle autre ? De même, il vaut mieux ne pas prévoir de travail écrit dans le train, etc.

Commentaires sur le programme

Le programme ici présenté sur un Tandy TRS-80 a pour configuration une mémoire de 4 K octets de mémoire vive MEV, un écran-clavier, un cassettophone. Le langage utilisé est le BASIC niveau II.

Le programme permet de prendre en compte 25 articles au minimum, dans une même période. (Nous appellerons *article* une plage horaire occupée sur l'agenda).

L'utilisateur peut choisir parmi six possibilités :

- création d'un article (avec contrôle des données entrées),
- correction d'un article,
- interrogation d'un groupe d'articles, basée sur une date de départ, le tri, avant la visualisation, est effectué à partir de cette date,
- liste de l'ensemble des articles : visualisation après tri, par page c'est-à-dire par groupe de 13 lignes,
- sauvegarde des articles : l'option FIN déclenche le transfert des articles en mémoire sur une cassette, ce qui constitue notre fichier,
- chargement du fichier-cassette en mémoire, avec sélection sur la date : ne sont retenues que les plages horaires postérieures à une date fixée.

Structure du programme

Le corps du programme se situe de 10 à 610 comme suit :

de 10 à 90 : identification de tableaux - impression des commandes possibles - branchements alternatifs (suivant option),

en 100 : création d'un article,

de 200 à 220 : liste fichier,

de 300 à 320 interrogation articles,

de 400 à 430 correction articles,

de 500 à 540. chargement en mémoire d'un éventuel fichier-cassette,

de 600 à 610 création du fichier-cassette (sauvegarde).

Nous trouvons ensuite six modules (ou sous-programmes):

de 1 000 à 1 099 vérification d'une date,

de 1 500 à 1 510 introduction d'une date,

de 2 000 à 2 020 introduction de l'article avec contrôle,

. *de 2 100 à 2 120* tri sur la date et l'heure de début des plages horaires,

en 2 300 impression de l'entête,

de 2 400 à 2 410. impression d'une ligne article, avec mise à jour du compteur de lignes.

Charger le programme

Lorsque le programme est chargé en mémoire, le message « READY » est imprimé,

taper au clavier « RUN » (puis enfoncer « ENTER »),

un menu présentant les six options est affiché l'utilisateur répond en sélectionnant un chiffre de 1 à 6

Messages divers :

DATE (SS/MM/AA) ? répondre de 01/01/79 à 31/12/80,

HEURE DE DEBUT ? répondre sous la forme 13.30,

HEURE DE FIN ? répondre sous la forme 13.30

Utilisation des fichiers-cassette

Chargement du fichier (option 5) DATE (SS/MM/AA) ? Ne seront chargés depuis la cassette que les articles postérieurs à cette date (comprise)

Sauvegarde sur cassette (option 6) rembobiner la bande cassette choisie pour le fichier, initialiser le compteur du magnétophone à zéro, puis, avant d'enfoncer les touches « RECORD et PLAY », faire avancer la bande de quelques blocs (compteur à 3 ou 4)

NB c'est une bonne idée que d'utiliser une cassette relativement courte (10 à 15 mn) et de changer de face à chaque utilisation de la cassette on a ainsi beaucoup plus de chances d'avoir au moins une version pas trop ancienne du fichier stocké. Mais il faut penser à marquer très lisiblement la date d'utilisation sur la bonne face de la cassette

Michel Favier

Liste du programme

```
10 REM PROGRAMME D'AGENDA
15 REM AUTEUR : MICHEL FAVIER
20 REM COPYRIGHT L'ORDINATEUR INDIVIDUEL ET L'AUTEUR
21 POKE 1+553,255  REM *** SPECIAL IRA-80 ***
25 CLEAR700:DIMJ$(26),O(26),F(26),O$(26) A$="TAPER  "
30 CLS
35 PRINT
40 PRINTA$,"1 SI CREATION ARTICLE"
45 PRINTA$,"2 SI DISP FICHIER"
50 PRINTA$,"3 SI INTERROGATION"
55 PRINTA$,"4 SI CORRECTION PRINT"
60 PRINTA$,"5 SI CHARGEMENT FICHIER"
65 PRINTA$,"6 SI FIN"
70 INPUT C$  CLS
75 IF C<1 OR C>6 THEN PRINT "UN CHIFFRE ENTRE 1 ET 6 SVP"  GOTO 35
80 ON C GOSUB 100  ,200  ,300  ,400  ,500  ,600
85 GOTO 30
90 REM -------------

100 REM - CREATION D'UN OU PLUSIEURS ARTICLES
110 FOR I=P+1 TO 26
120 GOSUB 1500  IF J$(I)="  THEN I=P+1  I=26  GOTO 140
130 GOSUB 2000
140 NEXT I
150 I=19
160 RETURN
200 REM - LISTE DES DONNEES
210 GOSUB2300  GOSUB2100  W=0
220 FOR I=1 TO I  GOSUB 2400  NEXT I
230 GOSUB 9100  RETURN
300 REM - INTERROGATION
310 I=0 GOSUB1500  GOSUB2300  GOSUB2100: W=0
320 FOR I=1 TO I  IF J$(I)=J$(0) THEN GOSUB 2410
330 NEXT I
340 GOSUB 9100  RETURN
400 REM - CORRECTION D'UN ARTICLE
410 I=0  GOSUB1500
420 FOR I=1 TO I
430 IF J$(0)<>J$(I) THEN 460  ELSE GOSUB 2300  GOSUB 2400
440 PRINT"MODIFICATION POUR CET ARTICLE (O OU N) "  GOSUB 9000
450 IF S$="O" THEN GOSUB 1520  GOSUB 2000
460 NEXT I
470 GOSUB 9100  RETURN
500 REM - CHARGEMENT EN MEMOIRE D'UN FICHIER CASSETTE
505 I=0 PRINT"PREPARER LE FICHIER-CASSETTE (PLAY)"
510 INPUT"DATE DE DEBUT (JJ/MM/AA)",J$(1) GOSUB1000
515 Z$=J$(1)  GOSUB 2500  J$=Z$  REM DATE SOUS FORME DE CHIFFRE
520 PRINT"CHARGEMENT ..." INPUT#-1,F
525 I=1 FOR J=1 TO F
530 INPUT#-1,J$(I)
535 INPUT#-1,P(I),F(I),O$(I)
540 Z$=J$(1)  GOSUB2500
545 IF Z$>=J$ THEN I=I+1  REM ON NE GARDE QUE LES DONNES DATES
550 NEXT J
555 F=I-1 RETURN
600 REM -- SAUVEGARDE SUR CASSETTE ET ARRET
610 PRINT PREPARER UNE CASSETTE VIERGE (PLAY ET RECORD)'
620 GOSUB 9100  PRINT' SAUVEGARDE...'  PRINT#-1,F
630 FOR I=1 TO F
640  PRINT#-1,J$(I)
650  PRINT#-1,O(I),P(I),O$(I)
660 NEXT I
670 PRINT' * TERMINE *"
680 END
690 REM -----------

990 REM -- VERIFICATION D'UNE DATE
1000 OK=1  IF LEN(J$(1))<>8 THEN 1040
1010 IF MID$(J$(1),3,1)<>"/" OR MID$(J$(1),6,1)<>"/" THEN 1040
1020 IF MID$(J$(1),1,2)<"01" OR MID$(J$(1),4,2)<"01" THEN 1040
1030 IF INSTR(J$(1),"0") AND INSTR(J$(1),2)<"32"  THEN 1060
1040 PRINTCHR$(28),CHR$(27),CHR$(27),".. ERREUR DATE..",
1050 OK=0  GOSUB 9100
1060 RETURN
1490 REM -- LECTURE D'UNE DATE
1500 CLS PRINT "PRESSER ENTER ET FIN"
1510 J$(I)= "  REM POUR PRESERVER L'UTILISATION DE ENTER
1520 INPUT"DATE (JJ/MM/AA)..",J$(I)
1530 IF J$(I)= "  THEN I=I-1
                ELSE GOSUB 1000  IF OK=0 THEN 1500
1540 RETURN
1950 REM -- INTRODUCTION DE L'ARTICLE ET CONTROLE
2000 INPUT "HEURE DE DEBUT. " O(I)
2010 IF O(I)<0 OR O(I)>24 THEN PRINT CHR$(27)  GOTO 2000
2020 INPUT "HEURE DE FIN. ", F(I)
2030 IF F(I)<0 OR F(I)>24 THEN PRINT CHR$(27)  GOTO 2020
2040 INPUT "OBSERVATIONS...",O$(I)  O$(I)=LEFT$(O$(I),25)
2050 RETURN
2090 REM -- TRI SUR DATE ET HEURE
2100 FOR I=P TO I STEP -1
2110 I=I  REI DONNES NON TRIEES DANS L'ORDRE
2120 FOR N=I+1 TO I STEP -1
2130      Z$=J$(N)  GOSUB 2500  Z9=Z9
          Z$=J$(N+1)  GOSUB 2500  Z9=Z9
          IF Z9<Z9 THEN 2200
2140      IF Z9>Z9 THEN IF O(N)<O(N+1) THEN 2200
2150      T=O REM LES DONNEES NE SONT PAS DANS L'ORDRE
2160      Z$=J$(N)  J$(N)=J$(N+1)  J$(N+1)=Z$
2170      Y=O(N)    O(N)=O(N+1)    O(N+1)=Y
2180      Y=F(N)    F(N)=F(N+1)    F(N+1)=Y
2190      Z$=O$(N)  O$(N)=O$(N+1)  O$(N+1)=Z$
2200 NEXT N
2210  IF I=1 THEN N=2  REM LES DONNEES SONT DANS L'ORDRE
2220 NEXT N
2230 RETURN
2300 REM -- IMPRESSION DE L'EN-TETE
2310 CLS
2320 PRINT"  DATE        DEBUT       FIN          OBSERVATIONS"
2330 RETURN
2390 REM -- IMPRESSION D'UNE LIGNE
2400 IF I-W>14 THEN PRINT "SUITE"  GOSUB 9100  GOSUB 2300  W=W+13
2410 PRINT J$(I),
2420 PRINT USING "  ##.## H    ## ## H.", O(I),F(I),
2430 PRINT TAB(34),O$(I)
2440 RETURN
2500 REM -- TRANSFORMATION D'UNE DATE EN UN NOMBRE
2510 REM CE SOUS-PROGRAMME TRANSFORME LA CHAINE Z$ CONTENANT
2520 REM UNE DATE SOUS LA FORME JJ/MM/AA EN UN NOMBRE Z9 QUI
2530 REM VAUT 10000*(AA-79)+100*MM+JJ
2540 REM CECI PERMET LES COMPARAISONS DE DATE
2550 Z9=10000*(VAL(RIGHT$(Z$,2))-79)
        + 100* VAL(MID$(Z$,4,2))
        +      VAL(LEFT$(Z$,2))
2560 RETURN
2570 REM ------------

9000 REM -- OUI OU NON
9010 INPUT S$  S$=LEFT$(S$,1)  IF S$= "O"  OR S$="N" THEN 9030
9020 PRINT"O OU N SVP",  GOTO 9010
9030 RETURN
9100 REM -- ATTENTE
9110 PRINT"  QUAND VOUS VOULEZ",
9120 R$=INKEY$  IF R$="" THEN 9120
9130 RETURN
```



```
GRILLE POUR  3 EQUIPES
R= 1 A/C  SE REPOSE : B
R= 2 B/C  SE REPOSE : A
R= 3 A/B  SE REPOSE . C
GRILLE POUR  5 EQUIPES
R= 1 A/E B/D  SE REPOSE . C
R= 2 B/E C/D  SE REPOSE . A
R= 3 C/E A/B  SE REPOSE : D
R= 4 D/E A/C  SE REPOSE : B
R= 5 A/D B/C  SE REPOSE : E
GRILLE POUR  7 EQUIPES
R= 1 A/G B/F C/E  SE REPOSE  D
R= 2 B/G C/F D/E  SE REPOSE  A
R= 3 C/G D/F A/E  SE REPOSE  B
R= 4 D/G E/F A/C  SE REPOSE  B
R= 5 E/G A/D B/C  SE REPOSE  F
R= 6 F/G A/E B/D  SE REPOSE . C
R= 7 A/F B/E C/D  SE REPOSE  G
GRILLE POUR  9 EQUIPES
R= 1 A/I B/H C/G D/F  SE REPOSE : E
R= 2 B/I C/H D/G E/F  SE REPOSE . A
R= 3 C/I D/H E/G A/H  SE REPOSE : F
R= 4 D/I E/H F/G A/C  SE REPOSE : B
R= 5 E/I F/H A/D B/C  SE REPOSE  G
R= 6 F/I G/H A/E B/D  SE REPOSE  C
R= 7 G/I A/F B/E C/D  SE REPOSE . H
R= 8 H/I A/G B/F C/E  SE REPOSE . D
R= 9 A/H B/G C/F D/E  SE REPOSE : I
GRILLE POUR  11 EQUIPES
```

clubs

droit au but : préparez le concours de votre club de boules

Un jour, nous avons reçu une lettre d'Ignace Nicolosi qui nous appelait au secours pour l'aider à écrire un programme de préparation de tournoi de boules. Nous retransmettions alors son appel en « Courrier des lecteurs » et grâce à vous, et en particulier grâce à Mohamed Belaroui, le tournoi pouvait se dérouler en septembre sur ordinateur.

Membre d'un club de boules et passionné d'informatique (depuis peu !), je voulais lier mes deux passions pour l'organisation du tournoi annuel. Ce tournoi met au prise X equipes et X peut être impair. Dans ce cas la une equipe est au repos a chaque tour tandis que les autres se rencontrent deux a deux. Avec X petit, écrire un programme qui précise le déroulement du concours est simple si l'on utilise la technique de la permutation circulaire. Mais comme il devait y avoir plus de 20 équipes, le probleme devenait trop complique pour être résolu ainsi. D'ou ma lettre à L'OI. Puis l'attente. Une longue attente jusqu a fin août ou enfin je reçus la visite de Mohamed Belaroui et de son programme salvateur (*). Il ne me restait qu'à adapter ce programme sur mon CBM Commodore et bien sûr à l'essayer. Quant au test en grandeur nature il n'aura lieu que dans quinze jours.

Ce programme est bien sûr fort simple et par la même ne doit pas necessiter de nombreuses modifications pour être adapté à l'organisation d'autres tournois.

Si vous ne disposez pas d'une imprimante sur votre Pet/CBM, il suffit de supprimer les lignes 100 et 110.

Ignace Nicolosi

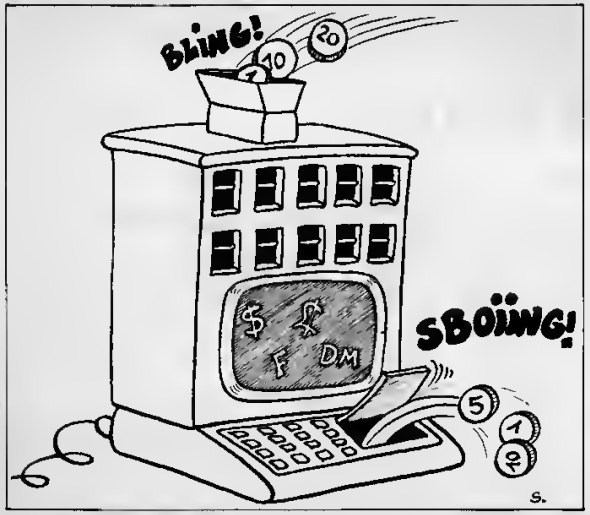
(*) Nous avons reçu d'autres programmes traitant ce probleme et remercions ici tous les lecteurs qui nous les ont envoyé. NDLR

```
10 REM   * PREPARATION DU TOURNOI DE BOULES *
20 REM  AUTEURS MOHAMED BELAROUI ET IGNACE NICOLOSI
30 REM  COPYRIGHT L'ORDINATEUR INDIVIDUEL ET LES AUTEURS
40 REM*****************************************************
100 OPEN 5,4            :REM POUR IMPRIMER
110 CMD 5
120 PRINT"
130 DIM T1(50),T2(25),R$(25),L$(25),R1$(1)
140 REM-----------
150 REM
160 FOR X=3 TO 25 STEP 2
170 PRINT TAB(10);"GRILLE POUR ";X;"EQUIPES"
180 PRINT"-----------------------------------------------
190 M=INT(X/2)
200 Y=X+(M*2)
210 IF Y=0 THEN 540
220 REM---
230 REM
240 FOR I=1 TO X STEP 1
250 READ L$(I): T1(I)=I
260 T1(I+X)=1: T2(I)=X-I+1
270 NEXT 1
280 REM
290 REM---
300 R=0
310 REM---
320 FOR I=1 TO X
330 R$(I)=" "
340 NEXT 1
350 REM
360 REM---
370 PRINT"R=",R+1;" ";
380 REM---
390 REM
400 FOR I=1 TO X
410 IF T1(I+R)=T2(I) THEN 460
420 IF R$(T1(I+R))="0" THEN 470
430 IF R$(T2(I))="0" THEN 470
440 PRINT L$(T1(I+R)); / ;L$(T2(I)),"  ";
450 R$(T1(I+R))="0":R$(T2(I))="0": GOTO 470
460 R1$=L$(T2(I))
470 NEXT I
480 REM
490 REM---
500 PRINT " SE REPOSE : ";R1$
510 PRINT"-----------------------------------------------
520 R=R+1
530 IF R<>X THEN 320
540 RESTORE
550 NEXT X
560 REM
570 REM-------
580 END
600 REM*****************************************************
1000 DATA "A","B","C","D","E","F","G","H","I","J"
1010 DATA "K","L","M","N","O","P","Q","R","S","T"
1020 DATA "U","V","W","X","Y"
```

Première parution : L'OI n° 22

enfin une valeur sûre pour gérer votre portefeuille de titres

Pourquoi ne pas vous aider de votre Sinclair ZX-81, muni de son extension 16 Ko MEV et éventuellement de l'imprimante, pour gérer votre portefeuille de titres ? Vous pourrez obtenir grâce à ce programme, outre la comptabilisation des valeurs détenues (jusqu'à trente valeurs différentes numérotées de 1 à 30), le calcul de la rentabilité globale de votre portefeuille, ainsi que celui de chaque valeur.



Au préalable, il vous faudra entrer en mémoire près de 9 K-octets, en prenant bien soin de décrire chaque ligne telle qu'elle figure dans la liste du programme, notamment pour tout ce qui est affichage de messages alphanumériques (même nombre d'espaces, etc.) et pour tout ce qui concerne la tabulation écran ou imprimante.

La toute première fois que vous utilisez le programme vous devez faire RUN, puis, après l'affichage du menu, vous devez nécessairement choisir l'option 5, c'est-à-dire la remise à zéro des fichiers mouvements.

Tapez « oui » en toutes lettres, cela est indispensable, car certaines variables doivent être déclarées au cours de cette portion de programme. Vous pouvez alors mettre à jour votre fichier valeur.

Vingt caractères sont disponibles pour chaque valeur

Pour les utilisations ultérieures (le programme ayant été sauvegardé précédemment par l'exécution de la ligne 9020 SAVE « BOURSE »), après chargement du programme contenant les données de votre portefeuille, l'exécution se poursuit automatiquement par un GOTO à la ligne 100.

Si au cours de vos transactions, il arrive que le programme s'arrête sur un compte rendu d'er-

Première parution : L'OI n° 44


```
1 POUR M.A.J. FICHIER VALEUR

2 POUR EDITION FICHIER VALEUR

3 POUR JOURNAL DES MOUVEMENTS

4 POUR EDITION D UN RELEVE

5 POUR REMISE A ZERO DES
  FICHIERS MOUVEMENTS

9 SI FIN
```

Menu général

```
QUEL N° VOULEZ METTRE A JOUR ?
(99 SI FIN)

NUMERO 1

LIBELLE  ESSAI

COURS  0 FRANCS

FRAPPER 1 POUR MODIFIER LE
          LIBELLE
        2 POUR MODIFIER LE COURS
        9 SI FIN

VOTRE REPONSE  2
```

Menus pour chaque option

```
1 POUR L IMPRIMANTE

2 POUR L ECRAN

9 SI FIN
```

```
1 POUR L IMPRIMANTE

2 POUR L ECRAN

9 SI FIN
```

Programme de gestion d'un portefeuille de valeurs (suite et fin)

```
3200 LET Q(N)=Q(N)+R
3205 LET M(N)=M(N)+T
3206 IF Q(N)=0 THEN LET PRU=0
3208 IF Q(N)=0 THEN GOTO 3220
3210 LET PRU=(INT (M(N)/Q(N)*100
))/100
3220 LET ACH=ACH+T
3225 GOTO 3500
3300 LET Q(N)=Q(N)-R
3305 LET M(N)=M(N)*Q(N)/(Q(N)+R)
3310 LET M(N)=(INT (M(N)*100))/1
00
3315 LET PVD=T-(R*PRU)
3320 LET VTE=VTE+T
3325 LET PVT=PVT+PVD
3330 GOTO 3500
3400 LET CPN=CPN+T
3500 PRINT AT 15,0,"SITUATION PO
STERIEURE"
3505 PRINT AT 16,0,"*************
*****"
3510 PRINT AT 17,0,"NOMBRE DE TI
TRES  ",Q(N)
3515 PRINT AT 18,0,"PRIX REVIENT
 GLOBAL  ",M(N)," FF"
3520 PRINT AT 19,0,"PRIX REVIENT
 UNITAIRE  ",PRU," FF"
3525 IF X=0 THEN PRINT AT 21,0,"
PLUS VALUES DEGAGEES  ",PVD," FF"
3528 PRINT AT 0,5,"
"
3530 COPY
3535 IF INKEY$="" THEN GOTO 3535
3540 GOTO 3010
4000 CLS
4005 PRINT TAB 7,'[redacted]
[redacted]'
4010 PRINT AT 7,0, 1 POUR L IMPR
IMANTE
4015 PRINT AT 10,0, 2 POUR L ECR
AN'
4020 PRINT AT 13,0, '9 SI FIN'
4025 IF INKEY$<>"1" AND INKEY$<>
"2" AND INKEY$<>"9" THEN GOTO 40
25
4030 LET R=CODE INKEY$-28
4035 IF R=9 THEN GOTO 100
4037 PRINT AT 16,0,"DATE D EDITI
ON?"
4038 INPUT X$
4040 IF R=1 THEN GOTO 4560
4100 CLS
4105 LET PTF=0
4110 LET U=0
4115 PRINT TAB 8,'[redacted]
[redacted]'
4120 PRINT TAB 6,'AV ",X$
4125 FOR N=1 TO 30
4130 IF Q(N)=0 THEN GOTO 4180
4135 LET MT=Q(N)*C(N)
4140 GOSUB 9100
4145 PRINT AT U+2,0,"*",U$(N)
4147 PRINT AT U+2,Y,MT
4150 PRINT Q(N), R ,C(N)," PF
4155 LET PRU=(INT (M(N)/Q(N)*100
))/100
4160 PRINT ,"PRU  ",PRU," PRG ",
M(N)
4165 PRINT ,"DER. OP. LE ",Q$(N)
(1 TO 2), ",D$(N)(3 TO 4),".",D
$(N)(5 TO 6)
4170 LET PTF=PTF+MT
4175 LET U=U+5
4176 IF U=20 THEN PRINT "APPUYER
POUR LA SUITE"
4180 IF U=20 THEN GOSUB 9998
4181 IF U=20 THEN PRINT AT 21,0
'                    "
4185 IF U=20 THEN LET U=0
4190 NEXT N
4195 LET MT=PTF
4210 GOSUB 9100
4220 PRINT AT 21,15, 'TOTAL ",TAB
Y,PTF
4230 GOSUB 9998
4235 GOTO 4000
4240 LPRINT TAB 8,'[redacted]
[redacted]'
4501 CLS
4503 LPRINT
4505 LPRINT TAB 6,"AV ",X$
4510 LPRINT
4515 LET PTF=0
4516 LET PRGT=0
4520 LPRINT
4525 FOR N=1 TO 30
4530 IF Q(N)=0 THEN GOTO 4590
4535 LET MT=Q(N)*C(N)
4540 GOSUB 9100
4545 LPRINT ,'*',U$(N),TAB Y,MT
4540 LPRINT ,'[redacted]'
4550 LPRINT Q(N),    R  ,C(N)," F
F"
4555 LET PRU=(INT (M(N)/Q(N)*100
))/100
4560 LPRINT ,"PRU  ",PRU,' PRG '
,M(N)
4565 LPRINT ,'DER. OP. LE ',Q$(N
)(1 TO 2), ",D$(N)(3 TO 4),".",
D$(N)(5 TO 6)
4567 IF M(N)=0 THEN GOTO 4570
4568 LET VAR=(MT/M(N)-1)*100
4569 LPRINT ,'VAR. ',(INT (VAR*1
00))/100,' PC'
4570 LET PTF=PTF+MT
4571 LET PRGT=PRGT+M(N)
4575 LPRINT
4590 NEXT N
4605 LET MT=PTF
4610 GOSUB 9120
4615 LPRINT
4620 LPRINT TAB 15, '[redacted]',TAB Y
,PTF
4625 LET MT=PRGT
4630 GOSUB 9100
4635 LPRINT
4640 LPRINT ,"PRIX REVIENT TOTAL
 TAB Y-2 ",TAB Y,PRGT
4645 LPRINT TAB 21,'------------
4650 LPRINT
4655 LET MT=PTF-PRGT
4656 LET DIFF=MT
4657 LET MT=ABS MT
4658 GOSUB 9100
4659 IF DIFF<0 THEN LET Y=Y-1
4685 LPRINT ,'VARIATION EN FF '
TAB Y,DIFF
4687 IF PRGT=0 THEN GOTO 4690
4688 LET VAR=(PTF/PRGT-1)*100
4689 GOSUB 9210
4690 IF PRGT=0 THEN GOTO 4690
4695 GOSUB 9100
4697 IF PVT<0 THEN LET Y=Y-1
4698 LPRINT ,"PLUS VALUES DEGAGE
ES  ",TAB Y,PVT
4693 IF PRGT=0 THEN GOTO 4705
4695 LET VAR=(PVT/PRGT)*100
4700 GOSUB 9210
4705 LET MT=CPN
4710 GOSUB 9100
4720 LPRINT ,"COUPONS ENCAISSES"
TAB Y,CPN
4722 IF PRGT=0 THEN GOTO 4735
4725 LET VAR=(CPN/PRGT)*100
4730 GOSUB 9210
4735 LPRINT
4740 LET MT=ABS (DIFF+PVT+CPN)
4745 GOSUB 9100
4746 IF DIFF+PVT+CPN<0 THEN LET
Y=Y-1
4750 LPRINT , TOTAL..........",T
AB Y,DIFF+CPN+PVT
4753 IF PRGT=0 THEN GOTO 4755
4755 LET VAR=((DIFF+CPN+PVT)/PRG
T)*100
4760 GOSUB 9210
4765 LPRINT
4770 LPRINT ,"TOTAL DES ENTREES
 ",ACH
4775 LPRINT ,"TOTAL DES SORTIES
 ",VTE
4780 LPRINT
4785 LPRINT ,"*******************
************"
4790 GOTO 4000
5000 CLS
5005 PRINT AT 8,0,'OUI OU NON VO
ULEZ VOUS PAIRE LA'
5010 PRINT AT 10,9, REMISE A ZER
O ?
5012 INPUT X$
5013 IF X$<>"OUI" THEN GOTO 100
5015 LET ACH=0
5020 LET VTE=0
5025 LET CPN=0
5030 LET PVT=0
5035 PRINT AT 12,0,'SI VOUS AVEZ
 MIS OUI AU LIEU DE NON...',,,,'
TOUT PIS C EST FAIT.'
5035 GOSUB 9998
5040 GOTO 100
9000 CLS
9005 PRINT AT 8,3, SAUVEGARDEZ V
OTRE FICHIER
9010 PRINT AT 12,0,'FRAPPEZ UNE
TOUCHE APRES AVOIR  PREPARE VOTR
E MAGNETOPHONE
9015 GOSUB 9998
9020 SAVE  "BOURSE"
9025 GOTO 100
9030 LET Z=1
9050 IF N>=10 THEN LET I=0
9070 RETURN
9080 LET Y=0
9090 IF C(N)>=10 THEN LET Y=1
9091 IF C(N)>=100 THEN LET Y=2
9092 IF C(N)>=1000 THEN LET Y=3
9093 IF C(N)>=10000 THEN LET Y=4
9094 RETURN
9100 LET Y=26
9101 IF MT>=10 THEN LET Y=27
9102 IF MT>=100 THEN LET Y=28
9103 IF MT>=1000 THEN LET Y=29
9104 IF MT>=10000 THEN LET Y=24
9105 IF MT>=100000 THEN LET Y=23
9106 IF MT>=1000000 THEN LET Y=2
2
9107 RETURN
9210 LPRINT ,"SOIT ",(INT (VAR*1
00))/100," PC"
9219 RETURN
9300 PRINT AT 15,0, 'ETE VENDUE )
 ETE POSSEDEE,  ' OPERATION RE
FUSEE,'
9395 GOSUB 9998
9310 GOTO 3012
9995 IF INKEY$=" " THEN GOTO 9990
9999 RETURN

                    9860 OCTETS DE PROGRAMME.
```

Liste des variables

A	Divers	PTF	Valeur du portefeuille
ACH	Total des entrées	PVD	Plus-value dégagée sur une opération
C (N)	Cours de la valeur de type N	PVT	Total des plus-values dégagées
CPN	Total des coupons encaissés	O (N)	Nombre de valeurs de type N détenues
D$ (N)	Date de la dernière opération sur la valeur de type N	R	Quantité négociée
DIFF	Différence entre prix de revient et valeur du portefeuille	S$	Sauvegarde provisoire du libellé code opération
E$	Date d opération	T	Montant d'une opération
M (N)	Prix de revient global de la valeur de type N	V$ (N)	Libellé de la valeur de type N
O$ (X)	Type d opération (entrée sortie, etc.)	VAR	Variation en pourcentage entre deux montants
P	Cours de negociation	VTE	Total des sorties
PRG	Prix de revient global d une valeur	W	Compteur de lignes lors d édition à l'écran
PRGT	Prix de revient global du portefeuille	X$	Divers
PRU	Prix de revient unitaire d une valeur	Y	Numéro de colonne pour édition
		Z	Utilisé pour la tabulation

reurs, un GOTO 100 en mode commande permet de reprendre l'exécution du programme, en affichant à nouveau le menu (surtout ne pas faire RUN).

Une fois le programme initialisé, procédez à la mise à jour du fichier valeur (option 1, vingt caractères sont disponibles pour le libellé de chaque valeur, le cours à mettre est celui de la date d'évaluation souhaitée).

Le programme de libellé des codes réalisé selon vos souhaits

Ce fichier valeur peut être alors visualisé sur écran ou imprimé par le choix de l'option 2 du menu. Pour l'impression, la date peut être entrée sous toutes les formes (par exemple : 31 août 1982).

Pour effectuer la saisie des opérations, prenez l'option 3 du menu. Vous disposez en tout de trois codes d'opérations :

Programme de gestion d'un portefeuille de valeurs (début)

```
AUTEUR    MR ALAIN KIENLEN

   5 REM SOURSE
  10 DIM V$(30,20)
  20 DIM D$(30,6)
  30 DIM O$(3,7)
  40 DIM Q(30)
  50 DIM M(30)
  70 DIM C(30)
  75 LET O$(1)= ENTREE
  77 LET O$(2)= SORTIE
  80 LET O$(3)= COUPON
 100 CLS
 101 PRINT TAB 13, [redacted]
 110 PRINT AT 4,0,'1 POUR M.A.J.
FICHIER VALEUR
 120 PRINT AT 7,0,'2 POUR EDITIO
N FICHIER VALEUR
 130 PRINT AT 10,0,'3 POUR JOURN
AL DES MOUVEMENTS
 140 PRINT AT 13,0,'4 POUR EDITI
ON D UN RELEVE
 150 PRINT AT 16,0,'5 POUR REMIS
E A ZERO DES  FICHIERS M
OUVEMENTS
 170 PRINT AT 20,0,'9 SI FIN'
 180 IF INKEY$()'1' AND INKEY$()
'2' AND INKEY$()'3' AND INKEY$()
'4' AND INKEY$()'5' AND INKEY$()
'9' THEN GOTO 100
 190 LET A=CODE INKEY$-28
 195 CLS
 200 GOTO 1000*A
1000 CLS
1001 PRINT TAB 3, [redacted]
[redacted]
1010 PRINT AT 4,0, QUEL NO UOULE
Z METTRE A JOUR ?  (99 SI FIN)'
1020 INPUT N
1025 IF N=99 THEN GOTO 100
1030 PRINT AT 7,0,'NUMERO ',N
1040 PRINT AT 9,0,'LIBELLE ',V$
(N)
1050 PRINT AT 11,0, COURS  ',C(N
) ' FRANCS
1060 PRINT AT 13,0,'FRAPPER 1 PO
UR MODIFIER LE  1 J
CELLE                           2 PO
UR MODIFIER LE COURS            2 SI
 FIN
1070 IF INKEY$()'1' AND INKEY$()
'2' AND INKEY$()'9' THEN GOTO 10
70
1075 LET A=CODE INKEY$-28
1080 IF A=0 THEN GOTO 1000
1085 PRINT AT 21,0, VOTRE REPONS
E  ,A
1090 IF A=1 THEN GOTO 1100
1095 IF A=2 THEN GOTO 1200
1100 INPUT V$(N)
1105 PRINT AT 21,0,
                    ,
1110 GOTO 1030
1200 INPUT C(N)
1205 PRINT AT 21,0,
                    "
1210 GOTO 1030
2000 CLS
2010 PRINT TAB 3, [redacted]
[redacted]
2020 PRINT AT 7,0, 1 POUR L INPR
IMANTE
2030 PRINT AT 10,0,'9 POUR L ECR
AN'
2040 PRINT AT 13,0, 9 SI FIN'
2050 IF INKEY$()'1' AND INKEY$()
'2' AND INKEY$()'9' THEN GOTO 20
50
2055 LET A=CODE INKEY$-28
2060 IF A=9 THEN GOTO 100
2000 IF A=2 THEN GOTO 2500
2005 CLS
2050 PRINT AT 10,0, INDIQUER LA
DATE
2025 INPUT X$
2100 LPRINT ,'*****************
**************
2110 LPRINT TAB 3,'EDITION DU FI
CHIER VALEUR
2115 LPRINT TAB 6,'AU ',X$
2117 LET Y$=
2120 LPRINT ,'*****************
**************
2130 LPRINT ,'CODE',TAB 10,'LIBE
LLE',TAB 20,'COURS'
2140 LPRINT
2150 FOR N=1 TO 30
2160 IF V$(N)#
     THEN GOTO 2190
2165 GOSUB 9050
2166 GOSUB 9090
2170 LPRINT TAB Z,N,"*",V$(N),"+
 ",TAB (28-Y),C(N)
2180 NEXT N
2185 LPRINT , 'FIN D EDITION'
2190 LPRINT ,'*****************
**************
2200 GOTO 2000
2500 CLS
2505 LET U=0
2510 PRINT TAB 3, EDITION DU FIC
HIER VALEUR
2520 PRINT ,'*****************
**************
2550 FOR N=1 TO 30
2560 IF V$(N)=
     THEN GOTO 2580
2565 LET U=U+1
2566 GOSUB 9000
2567 GOSUB 9050
2570 PRINT AT U+4,Z,N,"+",V$(N),
" ",TAB (28-Y),C(N)
2574 IF U=16 THEN GOSUB 9990
2575 IF U=16 THEN LET U=0
2580 NEXT N
2584 PRINT
2585 PRINT  FIN D EDITION"
2587 PRINT
2590 IF INKEY$="" THEN GOTO 2590
2600 GOTO 2000
3000 PRINT ,'*****************
**************
3005 LPRINT TAB 5, [redacted]
[redacted]
3010 LPRINT ,'*****************
**************
3012 CLS
3015 LET Z0=""
3020 PRINT AT 2,0,'CODE VALEUR?
(99 SI FIN)'
3025 INPUT N
3026 IF N=99 THEN LPRINT "FIN O
EDITION"
3027 IF N=99 THEN GOTO 100
3030 PRINT AT 2,0,V$(N),' CODE
 N
3035 PRINT AT 4,0, SITUATION PRE
CEDENTE DU ',D$(N)(1 TO 2),' ',D
$(N)(3 TO 4),' ',D$(N)(5 TO 6),
3040 PRINT AT 5,0, ***********
*******
3045 PRINT AT 8,0, QUANTITE DETE
NUE  ',Q(N)
3050 PRINT AT 7,0, PRIX REVIENT
GLOBAL  ',M(N)
3051 IF M(N)=0 THEN LET PRU=0
3052 IF M(N)=0 THEN GOTO 3060
3055 LET PRU=(INT (M(N)*D(N)*100
))/100
3060 PRINT AT 8,0, PRIX REVIENT
UNITAIRE  ',PRU
3065 PRINT AT 10,0,'CODE OPERATI
ON ?
3066 IF INKEY$()'1' AND INKEY$()
'2' AND INKEY$()'3' THEN GOTO 30
66
3070 LET X=CODE INKEY$-28
3075 PRINT O$(X)
3076 PRINT AT 11,0, 'AUTRE LIBELL
E ?
3077 INPUT L$
3078 IF L$="" THEN GOTO 3081
3079 LET S$=O$(X)
3080 LET O$(X)=L$
3081 PRINT AT 11,0,'DATE ?
3082 PRINT AT 21,0,'.'
3083 INPUT E$
3086 PRINT AT 21,0,' '
3087 IF LEN E$<8 THEN GOTO 3080
3090 PRINT AT 11,6,E$
3095 PRINT AT 10,0, QUANTITE ?
3100 INPUT R
3101 IF X=2 AND R>Q(N) THEN GOTO
3100
3105 PRINT AT 30,0,O$(X),' ',R,'
LE ',E$(1 TO 2),'.',E$(3 TO 4),
',E$(5 TO 8)
3106 IF S$()'' THEN LET O$(X)=S$
3110 PRINT AT 11,0, ***********
*******
3115 PRINT AT 12,0, AU COURS DE
3120 INPUT P
3125 PRINT P,' FF'
3130 PRINT AT 13,0,"MONTANT NET
3135 INPUT T
3140 PRINT T,' FF'
3145 PRINT AT 15,0,"SAISIE CORRE
CTE O/N ?
3146 FOR Z=1 TO 5
3147 NEXT Z
3148 PRINT AT 15,0,"[redacted]
[redacted]
3149 FOR Z=1 TO 5
3150 NEXT Z
3155 IF INKEY$()"O" AND INKEY$()
"N" THEN GOTO 3145
3160 IF INKEY$="N" THEN GOTO 301
0
3165 LET O$(N)=S$
3170 GOTO 3100+X*100
```

Copyright l Ordinateur individuel et Alain Kienlen

Programme d'édition d'un journal des mouvements

```
*********************************
[redacted]
*********************************

L ORD.INDIVIDUEL      CODE  1

SITUATION PRECEDENTE DU    .  .
**********************
QUANTITE DETENUE   0
PRIX REVIENT GLOBAL   0
PRIX REVIENT UNITAIRE  0

ACHAT   100 LE 12.06.82
**********************
AU COURS DE    280 FF
MONTANT NET   28253.37 FF

SITUATION POSTERIEURE
**********************
NOMBRE DE TITRES   100
PRIX REVIENT GLOBAL   28253.37 FF
PRIX REVIENT UNITAIRE   282.53 FF

*********************************

L ORD.INDIVIDUEL      CODE  1

SITUATION PRECEDENTE DU 12.08.82
**********************
QUANTITE DETENUE   100
PRIX REVIENT GLOBAL   28253.37
PRIX REVIENT UNITAIRE   282.53

ATTRIB. 10 LE 15.07.82
**********************
AU COURS DE    0 FF
MONTANT NET   0 FF

SITUATION POSTERIEURE
**********************
NOMBRE DE TITRES   110
PRIX REVIENT GLOBAL   28253.37 FF
PRIX REVIENT UNITAIRE   256.84 FF

*********************************

L ORD.INDIVIDUEL      CODE  1

SITUATION PRECEDENTE DU 15.07.82
**********************
QUANTITE DETENUE   110
PRIX REVIENT GLOBAL   28253.37
PRIX REVIENT UNITAIRE   256.84

COUPON  110 LE 30.07.82
**********************
AU COURS DE    8 FF
MONTANT NET   880 FF

SITUATION POSTERIEURE
**********************
NOMBRE DE TITRES   110
PRIX REVIENT GLOBAL   28253.37 FF
PRIX REVIENT UNITAIRE   256.84 FF

*********************************

L ORD.INDIVIDUEL      CODE  1

SITUATION PRECEDENTE DU 30.07.82
**********************
QUANTITE DETENUE   110
PRIX REVIENT GLOBAL   28253.37
PRIX REVIENT UNITAIRE   256.84

VENTE   40 LE 16.08.82
**********************
AU COURS DE   325 FF
MONTANT NET   12805.27 FF

SITUATION POSTERIEURE
**********************
NOMBRE DE TITRES   70
PRIX REVIENT GLOBAL   17979.41 FF
PRIX REVIENT UNITAIRE   256.84 FF

PLUS VALUE DEGAGEE  2531.67 FF
*********************************
FIN D EDITION
```

Programme d'un relevé de titres

```
[redacted]
AU 31 AOUT 1982

[redacted] ORD.INDIVIDUEL          24514

70  A 358.2 FF
PRU  256.84 PRG 17979.41
DER. OP. LE 16.08.82
VAR. 35.34 PC

[redacted]                         24614
PRIX REVIENT TOTAL              - 17979.41
                                ----------
VARIATION EN FF                     5534.89
SOIT 36.34 PC
PLUS VALUES DEGAGEES                2531.67
SOIT 14.08 PC
COUPONS ENCAISSES                    880
SOIT 4.89 PC

TOTAL.........                      9948.28
SOIT 65.32 PC

TOTAL DES ENTREES  28253.37
TOTAL DES SORTIES  12805.27

*********************************
```

les entrées : code 1,
les sorties : code 2,
les coupons : code 3.

Le programme vous demande cependant si vous souhaitez modifier le libellé correspondant à ces codes, par exemple « achat » au lieu d'entrée et « vente » au lieu de sortie. Le nouveau libellé ne doit pas comporter plus de six caractères. Si vous tapez seulement Newline, le libellé standard apparaît, par exemple : entrée.

Le ZX-81 est une valeur sûre, si vous savez l'utiliser !

La date, dont l'introduction est demandée à la suite du libellé, doit, cette fois-ci, être émise sous la forme JJ MM AA (par exemple : 3 mai 1982 = 030582).

Afin de ne pas fausser les fichiers « mouvements », il y a lieu, pour toute opération telle qu'une sortie de droits (qui n'est pas une vente) accompagnée d'une attribution d'action(s), de sortir les droits avec le code entrée et d'inscrire qualité et montant en « négatif ».

Il en est de même pour toute annulation d'opération enregistrée par erreur.

Ci-dessous, voici maintenant la décomposition du programme.

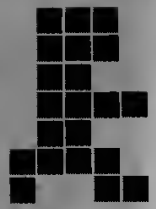
N'oubliez pas que votre Sinclair est une valeur sûre, mais qu'il ne fait que ce que vous lui demandez de faire.

Alain Kienlen

Programme

Lignes 10 à 80 :	initialisation des tableaux.
Lignes 100 à 200 :	affichage du « menu » général.
Lignes 1000 à 1210 :	mise à jour du fichier valeur.
Lignes 2000 à 2600 :	édition, sur l'écran ou sur l'imprimante, du fichier valeur.
Lignes 3000 à 3640 :	journal des mouvements, destiné à mettre à jour les quantités de titres détenus et à enregistrer les montants globaux des entrées, sorties et coupons encaissés.
Lignes 4000 à 4790 :	édition, sur l'écran ou sur l'imprimante, d'un relevé des valeurs détenues, indiquant différents renseignements.
Lignes 5000 à 5040 :	remise à zéro des fichiers mouvements, (entrées, sorties, coupons et plus values) à effectuer lors de la première utilisation du programme et à chaque fin d'année.
Lignes 9000 à 9025 :	sauvegarde du programme et des données de votre portafeuille. Vous y accédez par l'option 9 du menu général.

LE FESTIVAL DU LOGICIEL



Au cours du Festival du Logiciel qui a eu lieu à la Chartreuse de Villeneuve-lez-Avignon du 8 au 31 juillet 1983, le public a désigné les meilleurs des logiciels écrits par des auteurs "amateurs".

PALMARES

Grands prix décernés par le CIRCA La Chartreuse
Dominique Bertrand, pour son programme «Son» (10 000 FF)
Daniel Delevoye, pour son programme «Test de caractère» (10 000 FF)
Pascal Pellier, pour son programme «Formule 1» (10 000 FF)

Prix décerné par L'Ordinateur Individuel
Serge Nal, pour son programme «Damo» (10 000 FF)

Prix décernés par RTL
Grand prix «Jeune programmeur» à François Coulon, pour ses deux programmes «Aventure à Zorgon» et «Romulant» (7 000 FF)
Second prix «Jeune programmeur» à Christophe Blanchot, pour «La France et ses départements» (3 000 FF)
Prix «Club» au Club Ademir, pour «Krypton» (5 000 FF)

Prix décernés par Antenne 2 à cinq jeunes programmeurs
Eric Lablaine, pour son «Télécran» (2 000 FF)
Pascal Anquetin, pour «Venusia et l'empire des Elstres» (2 000 FF)
Gérard Samblancat pour «Master Mind» (2 000 FF)
Jean-Arthur Silve, pour «Galactic 3» (2 000 FF)
Jean-Luc Perrin, pour «Envahisseurs» (2 000 FF)

Prix décernés par l'Agence de l'Informatique
Premier prix à Gilbert Montagner, pour «La tête dans le plat» (une configuration Goupil 3)
Deuxième prix à Michel Perrault, pour son «Jeu de Kim» (une configuration TO 7)
Troisième prix a Stéphane de Luca, pour «Maestro» (TO 7)
Quatrième prix à Bertrand Gayet pour «Multigraph» (TO 7)
Cinquième prix à Jean Lépine, pour «Jeu de la pendule, encadrement» (TO 7)

Prix décernés par la Fondation de France
Premier prix à Jean Lépine pour «Labyrinthe et isométries» (10 000 FF)
Second prix à Pierre-Alain Goupille et Jean-Claude Robinet, pour leur programme «Unité Centrale» (5 000 FF)

Prix décerné par la Caisse des dépôts et consignations
Jean-Yves Garnier et Olivier Sidler, pour «Diabolo» (16 000 FF)

Prix décerné par la Compagnie Bull
Association française pour la lecture, pour «Elmo» (15 000 FF)

Prix décernés par Vifi-Nathan
Prix «Education» à Jean-Claude Attal, pour «Calcul Mental» (une configuration TO 7)
Prix «Jeu» à Philippe Baroin, pour «Descente à ski de l'Everest» (TO 7)
Prix «Gestion Familiale» à Dominique Otello, pour «Suivi de CCP» (TO 7)

Prix décerné par Thomson
William Hennebois, pour «Rebelle» (une configuration TO 7 complète)

Le Festival du Logiciel a été organisé par le CIRCA La Chartreuse avec le concours de la revue L'ORDINATEUR INDIVIDUEL et de la Coopérative de Conseils TEN, et avec la participation de RTL et d'Antenne 2

Référence 120 du service-lecteurs (page 172)

LE FESTIVAL DU LOGICIEL

Au cours du Festival du Logiciel qui a eu lieu à la Chartreuse de Villeneuve-lez-Avignon du 8 au 31 juillet 1983, le public non initié a utilisé plusieurs dizaines de logiciels écrits par des auteurs individuels ; une vingtaine d'autres logiciels étaient proposés par leurs éditeurs et sont disponibles dans le commerce.

Les premiers ont fait l'objet de prix dont leurs auteurs ont été lauréats, et les organisateurs sont heureux de mentionner ici les logiciels commerciaux qui ont été les plus demandés et appréciés par le public de ce Festival.

Galaxie-L : jeu d'adresse et de réflexion une navette spatiale dans un labyrinthe, a la recherche de diamants (édité par Ediciel (Matra-Hachette), rue Jean-Pierre Timbaud, BP 66 - 78390 Bois d'Arcy)

Pictor : logiciel pour composer à l'écran des tableaux réalistes ou abstraits, des paysages et pour les coloner (édité par Vifi Nathan, 17, rue d'Uzès 75002 Paris)

Diététique rappelle les principales notions et les grandes règles de la diététique puis passe à la pratique en composant un menu adapté aux besoins de l'utilisateur (edité par Vifi-Nathan)

EdiLogo : langage informatique interactif avec le graphisme de la tortue apprentissage de la logique du raisonnement et du travail en groupe (édité par Ediciel)

Léna 1 : un cours rédigé en français sous forme de 26 leçons de Basic Applesoft, 120 programmes expliqués et exécutés sur l'écran, et un jeu de 140 questions-réponses (édité par André Finot, 8 allée Bouffon, 91000 Evry-Courcouronnes)

Logo : langage de programmation pour tous, enfants et adultes n'ayant jamais touché à un ordinateur, permettant de créer des figures avec la tortue et le petit triangle sur l'écran couleur sans aucune instruction complexe (édité par SMT, 22 rue Saint-Amand, 75015 Paris)

Mélodia : qui permet de composer une mélodie, de l'écouter, de l'améliorer, de la transformer et d'apprendre quelques notions de solfège (édité par Vifi-Nathan)

Motus (jeu du cochon) le joueur propose une lettre à l'ordinateur, si elle fait partie du mot à trouver, elle s'affiche à sa place dans le mot, sinon le cochon perd une partie de son corps

Le Festival du Logiciel a été organisé par le CIRCA La Chartreuse avec le concours de la revue L'ORDINATEUR INDIVIDUEL et de la Coopérative de Conseils TEN, et avec la participation de RTL et d'Antenne 2

Référence 120 du service-lecteurs (page 172)

VISMO
informatique
22 boulevard de Reuilly 75012 Paris
Tél. 628.28.00

ORDINATEURS
Familiaux
et
Professionnels
Pour Tous

Pour obtenir un catalogue et la liste de nos prix, découpez ce bon :

Nom _____
Prénom _____
Adresse _____
_____ Ville _____
Code postal _____ Tél. _____
Date : Signature :

Référence 117 du service-lecteurs (page 172)

cf2e

EXPERTS EN LOGICIEL MICRO

Concepteurs, Réalisateurs, Distributeurs de progiciels
- Modules d'initiation E A O
- Applications familiales
- Jeux de réflexion et d'adresse

ORIC 1 — ORIC-1

- ☐ EDUSCOPE 7 - l'initiation au BASIC "colorée et sonore" 3 cassettes + manuel 30 pages — 360 F TTC
- ☐ ORISCOPE 1 - la maîtrise de la basse résolution 1 cassette + livret — 86 F TTC
- ☐ ORISCOPE 2 - la maîtrise de la haute résolution 1 cassette + livret — 86 F TTC
- ☐ ORISCOPE 3 - la maîtrise de la musique 1 cassette + livret — 86 F TTC
- ☐ COMPTASCOPE 2 - gestion familiale des recettes et dépenses — 160 F TTC

APPLE — apple

- ☐ EDUSCOPE 4 - l'initiation au BASIC APPLE SOFT 2 disques simples + manuel 70 pages — 590 F TTC

SPECTRUM

- ☐ Catalogue des logiciels "ARTIC" sur simple demande

ZX81 — ZX81

- ☐ EDUSCOPE 1 - initiation grand public au BASIC et concepts informatiques 4 cassettes - 1 manuel 100 pages — 360 F TTC
- ☐ EDUSCOPE 1 - 10 programmes d'applications - 1 cassette — 90 F TTC
- ☐ EDUSCOPE 2 - pour les passionnés - initiation à l'ASSEMBLEUR Z80 2 cassettes — 320 F TTC
- ☐ COMPTASCOPE 1 - gestion familiale - 1 cassette — 100 F TTC
- ☐ PLAYSCOPE 1 - le jeu des anagrammes - 1 cassette — 85 F TTC
- ☐ PLAYSCOPE 2 - 25 lettres au choix pour composer des mots "le compte est juste" - 1 cassette — 85 F TTC
- ☐ PLAYSCOPE 3 - OPLAH la grenouille espiègle — 85 F TTC

BULL MICRAL — Bull

- ☐ EDUSCOPE 6 - initiation à la programmation BASIC 1 disque souple + manuel 50 pages (préciser le type de disque souple à la commande D2 D4 D8) — 590 F TTC

SHARP MZ80A/B — SHARP

- ☐ EDUSCOPE 3 - initiation au BASIC 1 disque souple + manuel 85 pages — 540 F TTC

BON DE COMMANDE

Je désire recevoir les progiciels suivants indiqués dans les cases ci-dessus.
Je choisis de payer
☐ par CCP ou chèque bancaire, établi à l'ordre de CF2E, joint au présent bon de commande
☐ directement au facteur, moyennant une taxe de contre-remboursement de 16 F
Adresser vos commandes à
CF2E Val de Loire - 11 rue d'Alsace 49000 ANGERS - tél. (41) 88 62 76

Référence 116 du service-lecteurs (page 172)

VTR Software
54, rue Ramey 75018 PARIS

INTERCEPTEUR
MINEUR
DESTRUCTEUR
DINKEUR
CROQUEUR
EMBRASSEUR
TAMPONNEUR
ET BIEN D'AUTRES A VENIR

CARTOUCHES DISPONIBLES ACTUELLEMENT

DEMANDEZ NOTRE CATALOGUE GENERAL SOFTWARE CARTOUCHES ET CASSETTES pour ZX81, ORIC, SPECTRUM, VIC 20, COMMODORE 64

VOUS CONNECTEZ LA CARTOUCHE et VOUS JOUEZ !

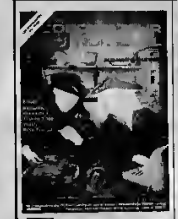
- CASES COMPUTER SIMULATIONS
- Mr. CHIP SOFTWARE
- HEWSON CONSULTANTS
- IMAGINE
- INTERCEPTOR MICRO'S
- J.K. GREY ENTERPRISES
- LLAMASOFT
- MARTECH GAMES
- MIKROGEN
- QUICK SILVA
- R and R SOFTWARE
- I.J.K. SOFTWARE
- NEW GENERATION SOFTWARE

Les produits de VTR Software sont disponibles dans les points de vente VTR Informatique.

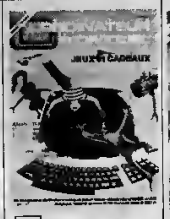
DEMANDE DE CATALOGUE
Nom : Prénom :
Adresse : Code Postal :

Référence 114 du service-lecteurs (page 172)

4 NUMÉROS POUR 66 FF*

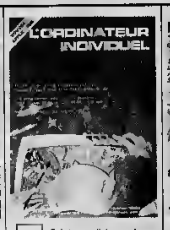


42 L'ordinateur et ses applications professionnelles : pourquoi et comment s'informatiser ? La facturation. Le traitement de texte. En archéologie. Dans une station-service. Dans un camping. La comptabilité et la trésorerie d'un chaussure ● Essais : Toshiba T 200, New Brain, Canon CX1, Hippomaxine GP 100, Vistelite le Rasco française ● Apprentissage de la lecture avec un MZ-80 ● Régler les files d'attente avec un ordinateur ● Comment faire parler votre ordinateur ● Jeux : guerre de l'espace sur ZX-81, faire du dessin animé ● Tournoi d'Othello ● Le SICOB 82, l'expo PCW de Londres ● Bien parler en Basic.



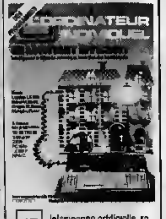
43 Jeux et cadeaux : une sélection de jeux électroniques ● Un ordinateur dans un atelier de menuiserie ● Un fichier de 1000 clients sur une disquette ● Essais : TI-59/4 A, Casio fx 601, Victor 2, Epson HC-20, ski sur Mattel, Kabul Spy, football sur TI-59/4 A, Wordindex, Othello, Visicalcul ● Enseignement et ordinateur ● Jeux : casse-briques, 421, train électrique, Awale, bridge, challenger contre bridge, duplicate ● La reconnaissance des formes ● à l'intérieur du FX-702 P du PC-1500 ●

44 Langages : lequel choisir ? Évolution des langages, le langage C, le langage Ada, interpréteurs et compilateurs, un système d'EAO ● Un apple dans une pyramide ● Une paroisse gérée par ordinateur ● Essais : PHC-8000 de Sanyo, Bus-80, Calcstar TO 7, ZX/M ● La gestion d'un portefeuille de titres sur un ZX-81 ● Musique sur clavier Qwerty ● Jeux : martingale, rubik's cube, billard ●



45 Création artistique et informatique : graphismes, musique et animation visuelle, interviews d'une artiste, enseignes des arts plastiques : tracer un trait, en quel : la bande dessinée graphisme sur TRS, apprendre la guitare, logo sur ZX-81 ● Les nouveautés de janvier ● Un ordinateur au Sahel ● Essais : Génie Couleur, Mattel Intellivision, Z 100, Armor Battle, Time Zone, Apple Plot, D-Base 2 ● Jeux : les naufragés sur ZX-81, les mêlionites sur Vic 20 ● Tout savoir sur e ●

46 Tout sur les ordinateurs portables et portatifs : panorama des 11 portables parlant Basic et des 20 portables que l'on pourra bientôt trouver en France, lequel pour votre examen ● Essais : Casio PB-100, CC-40, HP 75C, Apple 2e, Hustle, Apple-Writer 2, Silicon Office ● 17 000 01 dans un quartier : quel tel votre biorythme ? gérer des fichiers ● Jeux : échecs, tipper, labyrinthe ● Les codes-barres au Japon ● Un Basic en français ●

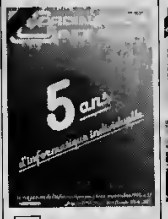


47 Intelligence artificielle robotique : insertion des handicapés : une maison autonome basée à Paris, des 01 pour handicapés moteurs ou sensoriels : la reconnaissance de la parole ● Essais : Dragon 32, Multisoft, Logabax LX 528, Bonzo, Pictor, Multiplan ● Dessiner ● Horloge sur TRS-80, nouveaux caractères sur Vic 20, géographie dans l'espace, oradles chinois sur ZX-81, lire la mémoire d'une HP-41 C ● Jeux : pétanque, Hector, Donjons et dragons ● Enquête : Lecteurs, qui êtes-vous ? ●

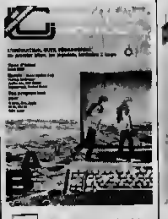
48 Les nouveaux outils du bureau : la bureautique, les 5 « calques » T/Maker, Supercalc, Calcstar, Multiplan, Vincalc... bien choisir une imprimante : les systèmes de traitement de texte et un programme sur CBM : modifier Wordstar ● Essais : extensions sur TI 99/4A, Oric 1, Alpsanie A8, Rainbow 100, Professional 350, Quiz Master, Homard internal, Blasto, Visidex ● La gestion d'un conseil municipal : apprendre la géographie ● Jeux : 24 Heures du Mans, alunissage, torpille et bathyscaphe ●

49 Les ordinateurs 16 bits : évolution technologique ou nouvelle mode ? : panorama de 01 16 bits : les processeurs 16/32 bits et leurs systèmes d'exploitation ● Essais : PC 8201, Goupil 3, Parsec, Calc4, and Get throats, Basicom ● L'informatique en milieu carcéral : le dessin animé, dessins en perspective, programme de tri rapide ● Jeux : envahisseurs ● Infora à Lyon, Cebit à Hanovre, West Coast Computer Fair, Micro-Expo ● Présentez correctement vos programmes ●

50 Les jeux : qu'est-ce que le jeu sur ordinateur ? 81 logiciels de jeux sur Apple 2, Atari 400, Atom, Sharp MZ 80, TI 99/4A, TRS 80, Vic 20, Victor Lambda, ZX 81 : programmez votre jeu ● Essais : Commodore 64, Sanyo 5001, Reporter et Philidor, Visischedule ● Programmation et jeux : gestion de cassettes sur TRS 80 : esquive sur HP 41 C, trésor des Templiers sur CBM, le pendu sur TI 59/4A ● Les jeux vidéo ● Expos : NCC et Apple Fest en Californie



51 5 ans d'informatique individuelle. En France : Sicob, Micro-Expo, aux États-Unis : NCC, au Japon : Micro Computer Show, en Belgique : Compec Europe 83 ● Essais : IBM PC, Lynx, Atari 400, AX 20, Factor, Alcor, Pascal Corp ● Programmation et jeux : distorsions de dessins sur ZX 81, musique et figures géométriques sur Apple 2, programmer plus vite sous CP/M en Z 80 (I) : les fantômes du Commodore 64 (I) : un explorateur sur Vic 20, Othello sur TRS 80 et IBM PC, casse-briques sur Pet de CBM ● Les banques de données



52 L'ordinateur outil pédagogique : les premiers bilans, se former pour travailler, s'initier à Logo (I), le langage Logo sur Atom, un problème de physique : des équations à géométrie variable, EAO, les trois didacticiels ● Essais : Mental 90/50, Victor Lambda 2HR, Jupiter Ace, Prestige Challenger, Calc-Result, Perfect Writer, Superscript ● Programmation et jeux : la géographie française sur Apple 2, rythme sur Oric, traitement de texte sur FX 702 P, programmez plus vite sous CP/M en Z 80 (II), les Zwarks à Oniopolis sur TRS, P Ti Man sur TI 99/4A, jeu de Yam avec un ZX 81, le vaisseau de la dernière chance sur CBM

53 19 ordinateurs d'un Basic à l'autre : Tableau comparatif pour adapter les programmes Basic d'un ordinateur à l'autre, non aux langages : choisir votre bibliothèque CP/M, programmer facilement en assembleur, programmation récursive sur Goupil et Apple 2 ● Alice, Gavilan ● Essais : Sharp MZ 730, Sanyo PHC 25, Tandy 100, Volkswriter, Omnis ● Programmation et jeux : horloge sur Vic 20 ou Oric, trajectoire sur PC 1500, programmez plus vite sous CP/M en Z 80 (III), les fantômes du Commodore 64 (II), shadocks sur ZX 81, l'Enterprise sur TRS 80 ● Expos : Sicob ● Sinclair à Logo (II) ■ Décoiffez des labels locaux.

BON DE COMMANDE

à retourner à L'ORDINATEUR INDIVIDUEL (Service Numéros), 39 rue de la Grange-aux-Belles, 75484 Paris Cedex 10 accompagné de votre règlement

Nom _____ Prénom _____

Adresse _____

Pays _____ code postal _____ Ville _____

Veuillez me faire parvenir les 4 numéros suivants (cochez les numéros choisis) :

42 43 44 45 46 47 48 49 50 51 52 53

* Ci-joint mon règlement de 66 FF (Étranger 80 FF, Belgique 535 FB, Suisse 22,5 FS)

Abonnez-vous à L'ORDINATEUR INDIVIDUEL

Pour être certain de ne manquer aucun numéro, **abonnez-vous.**
Vous recevrez tous les mois **L'ORDINATEUR INDIVIDUEL** chez vous
Et nous vous offrons **en plus** de faire une économie importante.

En nous retournant le bulletin-réponse ci-dessous,
vous paierez **198 FF** pour 11 numéros au lieu de **242 FF**
et vous recevrez **en cadeau** deux numéros à choisir à la page ci-contre.

BULLETIN A RETOURNER AUJOURD'HUI MÊME

à L'ORDINATEUR INDIVIDUEL service abonnements, 39 rue de la Grange aux Belles, 75484 Paris Cedex 10

Nom _____ Prenom _____

Adresse _____

Pays _____ Code postal _____ Ville _____

Veuillez m'abonner pour un an **(11 numéros)** à L'ORDINATEUR INDIVIDUEL
Par ailleurs je souhaite recevoir les **deux** numéros suivants (cochez les numéros choisis)

42 43 44 45 46 47 48 49 50 51 52 53

service lecteurs

Le service lecteurs de *L'Ordinateur Individuel* permet d'obtenir, des organismes et societes, des informations complementaires sur leurs activités et sur leurs produits. Les informations contenues dans la rubrique *Quoi de neuf* du *Magazine de l'informatique pour tous* sont referencees dans l'index ci-contre ; celles des publicites dans l'index ci-dessous. Utilisez la carte reponse page 11 en cerclant les références des informations, redaction ou publicite, qui ont retenu votre attention.

SL	Société	Page	SL	Société	Page
109	Becy	63	103	Micro expansion	174
116	CF2E	168	119	PME Computer	172
104	Esc Marseille	10	106	PSI Diffusion	8 et 9
112	Hercet micro informatique	100	113	Run Informatique	100
101	La commande electronique	2	115	Saan	14
118	La librairie informatique d aujourd hui	172	110	Semaphore Consultant	89
105	La Scool	6 et 7	117	Vismo	168
107	L ordinateur de poche	3	114	Vidéo Telemate Report	169
102	L ordinateur personnel	173	111	ZH Computer	92

Service librairie

La collection complète, les anciens numeros et les dernieres parutions de

L'ORDINATEUR INDIVIDUEL

sont disponibles à la

LIBRAIRIE INFORMATIQUE D'AUJOURD'HUI

253, rue Lecourbe, 75015 Paris - Metro Convention ou Boucicaut, ouvert du lundi au samedi de 9 h a 19 h

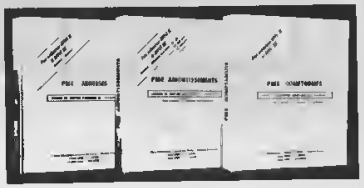
Librairie Informatique d'Aujourd'hui

tous vos livres et toutes vos revues

Pour toutes précisions sur la société ou le produit présenté ci-dessus : Référence 118 du service-lecteure (page 172)

LA MICRO-INFORMATIQUE AU SERVICE DES PME

apple apple apple



apple apple apple

Système d'exploitation : Ordinateur APPLE - 2 Floppy - Une imprimante série ou parallèle.

PME ADRESSES *
- 500 adresses par disquette
- Nombre de disquettes illimité
- Mise à jour du fichier adresses à chaque création
- Consultation instantanée des adresses.

A NOTER :
- Divers modes de sélection des étiquettes adresses et du classement
- Fichier adresses avec No de téléphone et nom du décisionnaire
- Mailing personnalisé avec le traitement de texte EPISTOLE

PME AMORTISSEMENTS *
- 500 immobilisations par disquette
- Nombre de disquettes illimité
- Mise à jour du fichier immobilisations à chaque création
- Consultation instantanée des immobilisations.

A NOTER :
- Calcul des plus ou moins-values à long et court terme.
- Etiquettes auto-adhésives pour identification physique des immobilisations.
- Changement ancien / nouveau Plan Comptable sans modification des saisies antérieures.

PME COMPTABILITE *
- 2500 écritures par 2 disquettes
- Nombre de disquettes illimité
- Mise à jour des comptes à chaque saisie d'écriture
- Consultation instantanée des comptes

A NOTER :
- Incorporation instantanée d'une écriture omise dans n'importe quel mois de l'exercice
- Bilan et compte d'exploitation simplifié
- Ancien et nouveau Plan Comptable.

Option disque dur en mono ou multiposte. Documentation sur demande et démonstration.

PME COMPUTER S.A. - 18, rue Botzaris 75019 PARIS - Tél. : 241.66.11

* MARQUE DEPOSEE

Pour toutes précisions sur la société ou le produit présenté ci-dessus : Référence 119 du service-lecteurs (page 172)

Imprimerie Bayard Presse 3-5 rue Bayard Paris 75008 Printed in France Photogravure Compo Relais 18 rue Le Peletier 75009 Paris
Dépôt légal imprimeur décembre 1983 Directeur de la Publication Jean-Luc Verhoye Directeur délégué Jean-Pierre Nizard
Diffusion NMPP Numéro de Commission paritaire 61042

Le guide des nouvelles solutions informatiques

L'ORDINATEUR PERSONNEL

Indispensables :
10 mots-clé pour faire la synthèse de l'année
Banques : la stratégie du cuirassé

Surprenantes :
les premières éliminatoires de notre coupe 83/84
Bancs d'essai : la crème des logiciels (2)
Des révélations sur Atari

Tout, tout, tout, sur les disquettes

Sévère mais juste : la sélection du Père Noël

N° 3

N° 3

20 FF chez votre marchand de journaux

Pour toutes précisions sur le société ou le produit présenté ci-dessus : Référence 102 du service-lecteurs (page 172)

Disques durs

MICRO-EXPANSION

5 Place Mal Lyautey [illegible] Tél. [illegible]

La Référence!!

capacités de 5 à 20 mégas avec sauvegardes

[illegible] toutes précisions sur la société ou le produit présenté [illegible]